



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Plataforma de comunicación basada en GPRS para administración de datos en
sistemas de automatización para invernaderos

Tesis

Que como parte de los requisitos para
obtener el grado de

INGENIERO EN AUTOMATIZACIÓN

Presenta:

Carlos Alberto Torres Ortega

Dirigida por

Dr. Genaro Martín Soto Zarazúa.

C.U. Santiago de Querétaro, Qro. Septiembre 2011

RESUMEN

Para poder tener un mejor control y actuación sobre un invernadero automatizado se ha creado una herramienta que administra de manera sencilla para los usuarios que la requieran todos los datos que involucran dicho invernadero, tales datos como la humedad, la temperatura y algunas variables climáticas, todo ello a través de la tecnología *General Packet Radio Service (GPRS)*. Los invernaderos se encuentran ubicados en el Campus Amazcala del municipio del Marques, Querétaro, México; punto estratégico debido a que se encuentra en una zona de gran actividad primaria y en donde se encuentran potenciales clientes para poder ofrecer los servicios de administración de datos vía GPRS. Para poder llevar a cabo el manejo de los informes que da el invernadero, se divide en dos partes el producto en una primera instancia tenemos al modem principal de comunicación GSM862-GPS de la marca TELIT, con el cual se hace comunicación para la recepción de los datos vía GPRS, en una segunda parte se tiene una tarjeta electrónica con un microcontrolador quien es el que decide que acción se tiene que realizar y está en constante comunicación con el modem GSM862-GPS, además este microcontrolador tiene esclavos que le indican o le envían la información que se está generando en los invernaderos cuando este lo requiera, en una tercera parte contamos con una interfaz gráfica diseñada para la computadora, es decir a partir de un software desarrollado con el compilador de Microsoft Visual C# se puede comunicar el usuario con los invernaderos de una forma sencilla, este programa se ha creado para enviar y recibir datos; y por último en la cuarta parte contamos con la interfaz grafica diseñada para los celulares, herramienta importante para que el usuario en cualquier parte donde se encuentre pueda realizar alguna acción si es necesario del invernadero que esta administrando o en cualquier caso pedir información del mismo.

DEDICATORIAS

Este proyecto de investigación quiero dedicarlo especialmente a mi papa, a mi mama y a mi hermana, quienes me han apoyado en toda mi vida.

AGRADECIMIENTOS

Para mí la parte medular ha sido mi familia, mi papa, mi mama y mi hermana, sin el apoyo de ellos y el ánimo constante que me entregaban a diario no lo hubiera hecho posible, además quiero agradecer el apoyo continuo de mis maestros quienes me dieron las herramientas, conocimiento y en ocasiones los materiales necesarios para realizar este trabajo; también a mis amigos que me apoyaron en múltiples tareas.

ÍNDICE

Resumen	2
Dedicatorias	3
Agradecimientos	4
Índice	5
I. INTRODUCCION	
1.1 Antecedentes	8
1.2 Descripción del problema	8
1.3 Justificación	10
1.4 Objetivo	11
1.5 Hipótesis	11
II. REVISION DE LITERATURA	
2.1 General Packed Radio Service (GPRS)	12
2.1.1 Arquitectura GPRS	12
2.1.1.1 GPRS Nodo de Soporte (SGSN)	14
2.1.1.2 GPRS Nodo de Soporte Gateway (GGSN)	15
2.1.1.3 Paquete de Unidad de Control (UCP)	15
2.2 Modulación del GPRS	15
2.3 GPRS Categorías de desempeño	16
2.4 Protocolos del plano de transmisión	16
2.4.1 GPRS Protocolo Tunneling (GTS)	17
2.4.2 Protocolo Convergencia Dependiente Subred (SNDCP)	17
2.4.3 Interferencia de aire	17
2.4.4 Capa de transferencia de datos	18
2.4.5 Capa física	18

2.5	Packet Data Protocol (PDP)	19
2.6	Acces Point Name (APN)	19
2.7	Point to Point Protocol	20
2.8	Socket	21
2.9	Firewall	21
2.10	Protocolo de Transmisión de Archivos (FTP)	22
2.11	Protocolo de Control de Transmisión (TCP)	22
2.12	Internet Protocol	23
2.13	HTTP Cliente y Servidor	24
2.14	Global System Mobile (GSM)	25
2.15	Global Position System (GPS)	25
2.16	Invernaderos	26
2.17	Microcontroladores	27
2.18	Software	28

III. METODOLOGIA

3.1	Ubicación del área del proyecto	29
3.2	Especificaciones de las instalaciones del equipo GPRS	30
3.3	Instalación y especificaciones del equipo	31
3.4	Descripción del módulo GM862-GPS TELIT	32
	3.4.1 Comunicación con el módulo GM862-GPS TELIT	32
	3.4.2 Comandos AT necesarios a utilizar	39
	3.4.3 Descripción de la tarjeta electrónica de comunicación	42
	3.4.4 Descripción de la tarjeta electrónica GM862-GPS TELIT	44
3.5	Desarrollo del software de para la administración de datos	47
	3.5.1 Programación del Microcontrolador	47
	3.5.2 Programación del software PC	54

3.5.3 Programación del software para un móvil	73
IV. RESULTADOS Y DISCUSIÓN	
4.1 Software	78
4.2 Hardware	89
4.3 Conclusiones y desarrollos futuros	90
LISTA DE ACRÓNIMOS	91
LISTA DE COMANDOS	94
LITERATURA	97
ANEXOS	99

I. INTRODUCCIÓN

1.1 ANTECEDENTES

Actualmente, las transmisiones inalámbricas constituyen una eficaz herramienta que permite la transferencia de voz, datos y vídeo sin la necesidad de cableado. Esta transferencia de información es lograda a través de la emisión de ondas de radio teniendo dos ventajas: movilidad y flexibilidad del sistema en general.

Las tecnologías inalámbricas han adoptado con el paso del tiempo una manera más sencilla y cómoda de utilizar toda clase de dispositivos con el fin de mejorar el confort y las comunicaciones en general, por ejemplo unir la tecnología en redes malla ZigBee con unidades GPRS para gestión de información.

ZigBee es un estándar para comunicaciones inalámbricas para soluciones que pueden ser implementadas por cualquier fabricante [2]. *ZigBee* es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radios digitales de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y bajo consumo de energía además de su fácil integración lo reafirma la viabilidad de integrar estas tecnologías para el trabajo propuesto.

1.2 DESCRIPCIÓN DEL PROBLEMA

La agricultura sigue siendo una de las actividades más importantes para la economía mundial, pues de ella depende la alimentación primaria de millones de personas. El clima, la riqueza del suelo, lo mismo que la extensión territorial, son factores determinantes para la capacidad productiva de un país, sin embargo, también influyen drásticamente la demanda de esos productos, la competencia de precios y la infraestructura que se crea alrededor para hacer rentable la producción [1].

En México, pese a tener una gran extensión territorial, clima y suelo fértil, hay otros elementos adversos que han producido una drástica modificación en la capacidad productiva. Uno de estos elementos es la limitación del desarrollo de tecnología, la cual por lo general es desarrollada por las grandes potencias, elevando de manera considerable los costos en la adquisición de alguna herramienta tecnológica de este tipo, dejando a los productores en un desbalance total cuando se habla de competencia, es por ello que se tiene un gran interés en el desarrollo de recursos tecnológicos que nos den el impulso competitivo que necesitamos a precios accesibles y de gran calidad para cubrir las diversas necesidades. Con el uso de ciencias aplicadas a esta rama, se incrementa la calidad en la producción con el uso adecuado de materias primas así como la cantidad energética necesaria, cuidando de esta forma el impacto y desgaste a los sistemas ecológicos.

Para poder generar tal calidad se tienen sistemas automatizados para invernaderos que nos brindan toda la confianza de tener siempre la misma calidad en la producción. Estos sistemas inteligentes utilizan constantemente el monitoreo de diversas variables, con el objetivo de extraer información relevante del medio para poder iniciar acciones con el fin de mantener la uniformidad en ambiente de manera automática, por mencionar ejemplos, en la Universidad Autónoma de Querétaro de la Facultad de Ingeniería se ha desarrollado una amplia cartera de sistemas de automatización para invernaderos. Para algunos casos se necesita la intervención del ser humano; el invernadero puede generar una alarma en el gabinete de control, en el mejor de los casos es inmediato el observar dicha alarma y aplicar las acciones requeridas para el proceso.

El trabajo de esta tesis propone el uso de la tecnología de los sistemas GPRS generando una plataforma de comunicación para administrar los datos arrojados por el control inteligente de los invernaderos, haciendo posible su visualización en una interfaz gráfica en el celular, para que el usuario, desde el lugar donde se encuentre pueda tomar las mejores decisiones en el control del invernadero. Los datos serán colectados por medio de una unidad maestra con tecnología ZigBee ya desarrollada.

1.3 JUSTIFICACIÓN

El uso indispensable de los celulares ha hecho de esta tecnología la mejor cuando se habla de comunicación. Algunas veces solo los utilizamos para escribir un mensaje de texto, hacer llamadas y enviar algunas fotos, podemos explotar la tecnología que utilizan para la comunicación de sus aplicaciones en la administración de un invernadero automatizado, pues nos proporcionan grandes ventajas cuando requerimos saber lo que está ocurriendo a una gran distancia de nuestra localización por ejemplo:

- 1) Alertar por falta de corriente eléctrica.
- 2) Dar a conocer las condiciones climáticas.
- 3) Alertar cualquier anomalía del controlador.
- 4) Alertar si ocurre algún incendio.
- 5) Dar un informe si es requerido, de la situación actual de control.
- 6) Avisa si alguna materia prima hace falta.
- 7) Activar bombas de riego.
- 8) Pedir que cualquier sensor de información.
- 9) Alerta si se está desarrollando una mayor producción de gases.
- 10) Poder tomar acciones desde cualquier lugar donde esté ubicado el usuario.

Un claro ejemplo de la potencialidad de esta herramienta; si el usuario no se encuentra cerca del invernadero pues ha sido llevado algunas hectáreas más allá a revisar la calidad del maíz, y ocurre un evento inesperado en el invernadero; el control inteligente del invernadero a través de la plataforma de comunicación GPRS mandará una señal de información pidiendo la acción inmediata para corregir tal evento, el usuario evitará tiempos de traslación que se traducen en la menor pérdida de materia, tomando decisiones correctivas desde el lugar donde se encontrase.

La plataforma de comunicación está formada por módulos inalámbricos, donde podemos interconectar diversos sistemas ya sean para el monitoreo de diversas variables como son temperatura, humedad, pH o CO₂ entre otras variables que nos dan a conocer las condiciones en las que está operando el invernadero.

El tener un sistema modular, tiene la ventaja que al tratarse grandes distancias que poseen algunos de los invernaderos, el ahorro en la instalación eléctrica y cableado de algún sistema de monitoreo resulta un costo extra que puede evitarse utilizando tecnologías inalámbricas y además el uso de energías alternativas como es el uso de celdas solares, le proporcionarían la fuente de alimentación que el sistema requiere.

1.4 OBJETIVOS

OBJETIVO:

Desarrollar una plataforma de comunicación con base en los sistemas GPRS mediante la administración de datos con la interfaz grafica del usuario (GUI) en celulares para monitoreo y control de sistemas automáticos para invernaderos.

1.5 HIPÓTESIS

HIPOTESIS:

Usando una plataforma de comunicación basada en la tecnología GPRS se administrará en mejor medida y con mayor rapidez en tiempo real información para controlar y monitorear los sistemas automatizados en invernaderos, desde cualquier parte donde se localice el usuario a través de una interfaz gráfica (GUI) instalada en un celular.

II. REVISIÓN DE LITERATURA

2. 1 General Packet Radio Service (GPRS)

El Servicio General de Paquetes por Radio o GPRS por su acrónimo en inglés (General Packet Radio Service) es una nueva tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de 'paquetes'. La conmutación de paquetes es un procedimiento más adecuado para transmitir datos, hasta ahora los datos se habían transmitido mediante conmutación de circuitos, procedimiento más adecuado para la transmisión de voz [3].

GSM es la tecnología de segunda generación más exitosa celular, pero la necesidad de mayores velocidades de datos dio lugar a nuevos avances para que los datos sean transferidos a un ritmo mucho más alto. El primer sistema para hacer un impacto en el mercado fue GPRS. El GPRS letras corresponden a General Packet Radio System, la tecnología GPRS a habilitado unas tasas de datos que se transmiten por una red celular en comparación con GSM.

GPRS se convirtió en el primer escalón del camino entre las redes GSM de segunda generación en tecnología celular y el 3G W-CDMA / UMTS. Con la tecnología GPRS los datos que ofrece servicios con tarifas de datos hasta un máximo de 172 kbps, instalaciones como la navegación web y otros servicios que requieren la transferencia de datos se hizo posible. A pesar de que algunos datos pueden ser transferidos a través de GSM, la tasa fue demasiado lento para aplicaciones de datos en tiempo real [4].

2.1.1 Arquitectura GPRS

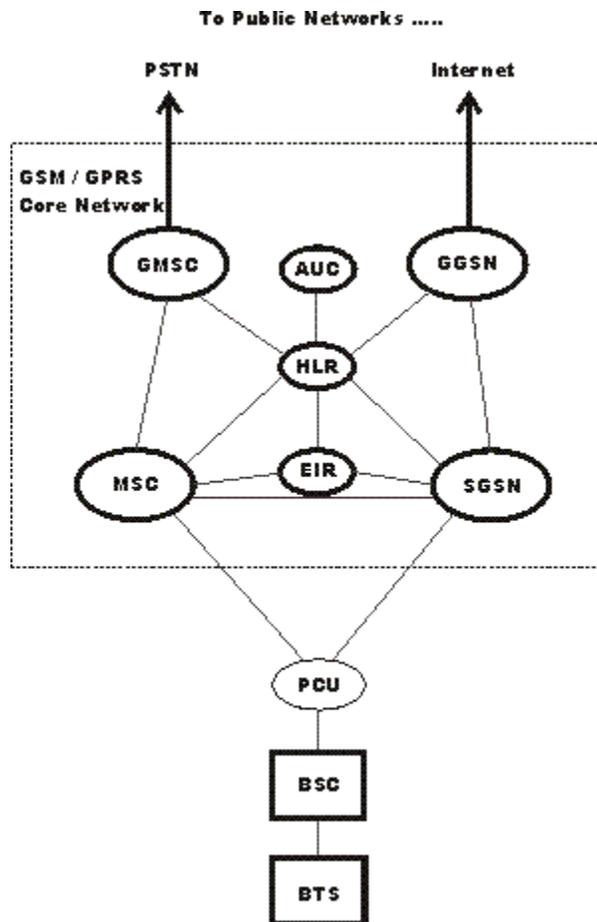
Con GPRS proporciona conectividad adicional en términos de paquetes de datos, hay, naturalmente, una serie de mejoras necesarias para la arquitectura de red utilizada. Una serie de nuevos elementos son necesarios para la red, pero puede funcionar junto con los elementos existentes en el sentido de que la capacidad

GPRS es una actualización a la red y no una estructura de red completamente nueva.

Las principales entidades en nueva arquitectura de red que se necesitan son:

- *SGSN*: GPRS Support Node - esta forma una puerta de acceso a los servicios dentro de la red.
- *GGSN*: GPRS Gateway Support Node que constituye la puerta de entrada al mundo exterior.
- *UCP*: Unidad de control de paquetes que diferencia si los datos se enrutan a la conmutación de paquetes o redes de conmutación de circuitos.

Una visión simplificada de la arquitectura de la red GPRS se puede ver en la esquema 1. Se observa que es muy similar a la arquitectura de red GSM más básico, pero con elementos adicionales.



Esquema 1. Arquitectura de la red GPRS

2.1.1.1 SGSN

El SGSN o porción nodo de soporte de GPRS, es elemento de la red GPRS que ofrece una serie de tomas, se centra en los elementos de propiedad intelectual de todo el sistema. Proporciona una variedad de servicios para los móviles:

- El enrutamiento de paquetes y la transferencia
- Gestión de la movilidad
- Conexión / desconexión
- Lógica de gestión de enlace
- Autenticación
- La carga de datos

Hay un lugar en el registro de esta información SGSN y los puntos de venta (por ejemplo, la celda actual, VLR actual). También almacena los perfiles de usuario (por ejemplo, IMSI, paquete de direcciones utilizadas) para todos los usuarios registrados GPRS con el SGSN en particular [4].

2.1.1.2 GGSN

El GGSN, GPRS Gateway Support Node es una de las entidades más importantes dentro de la arquitectura de la red GPRS.

El GGSN organiza la interoperabilidad entre la red GPRS y redes de conmutación de paquetes externos a los que los móviles se pueden conectar. Estos pueden incluir tanto a Internet y las redes X.25.

El GGSN puede ser considerado como una combinación de una puerta de acceso, router y firewall, ya que oculta la red interna hacia el exterior. En la operación, cuando el GGSN recibe datos dirigidos a un usuario específico, comprueba si el usuario está activo, el reenvío de los datos. En la dirección opuesta de paquetes de datos desde el móvil se dirige a la red de destino junto al GGSN.

2.1.1.3 UCP

La UCP o del paquete Unidad de Control es un router de hardware que se añade a la BSC. Se diferencia de datos destinado a la red estándar GSM (datos conmutados por circuito) y los datos destinados a la red GPRS (conmutación de paquetes de datos). La propia UCP puede ser una entidad física independiente, o con mayor frecuencia en estos días, se ha incorporado al controlador de estación base BSC, de tal modo de ahorro de los costes adicionales de hardware.

2.2 Modulación del GPRS

GPRS se basa en la estructura de base GSM. Utiliza el formato de la señal mismo ancho de banda que tiene 200 kHz. También tiene el mismo esquema de

modulación y el uso de la modulación GMSK. Conservando el mismo esquema de modulación, significa que el nivel de actualización necesario para ser capaz de soportar GPRS, además de GSM se reduce al mínimo.

2.3 GPRS Categorías de desempeño

No todos los móviles GPRS están diseñados para ofrecer los mismos niveles de servicio. Como resultado, se dividen en tres categorías básicas de acuerdo con sus capacidades en términos de la capacidad de conectarse a los servicios GSM y GPRS:

1. *Clase A:* - Esta clase describe los teléfonos móviles que pueden conectarse tanto a los servicios GSM y GPRS al mismo tiempo.
2. *Clase B:* - Estos móviles se puede conectar tanto a los servicios GPRS y GSM, pero que se puede utilizar en un solo servicio a la vez. Clase móvil puede hacer o recibir una llamada de voz o enviar y / o recibir un mensaje de texto durante una conexión GPRS B. Durante las llamadas de voz o mensajes de texto al servicio GPRS está suspendida pero se restablece cuando la llamada de voz o una sesión de SMS se ha completado.
3. *Clase C:* - Esta clasificación cubre los teléfonos que se pueden conectar a cualquiera de los servicios de GSM o GPRS, pero el usuario tiene que cambiar manualmente entre los dos tipos diferentes.

2.4 Protocolos del plano de transmisión

El plano de transmisión es el encargado de proveer la transmisión de los datos del usuario y su señalización para el control de flujo, detección de errores y la corrección de los mismos.

2.4.1 GPRS Protocolo Tunneling (GTS)

Es el encargado de transportar los paquetes del usuario y sus señales relacionadas entre los nodos de soporte de GPRS (GSN). Los paquetes GTP contiene los paquetes IP o X.25 del usuario. Por debajo de él, los protocolos estándares TCP o UDP se encargan de transportar los paquetes por la red. Resumiendo, en el Backbone del GPRS tenemos una arquitectura de transporte IP/X.25-sobre-GTP-sobre-UDP/TCP-sobre IP.

2.4.2 Protocolo Convergencia Dependiente Subred (SNDP)

Es el encargado de transferir los paquetes de datos entre los SGSN (nodo responsable de la entrega de paquetes al terminal móvil) y la estación móvil. Las funciones que desempeña:

- Multiplexación de diversas conexiones de la capa de red en una conexión lógica virtual de la capa LLC.
- Compresión y descompresión de los datos e información redundante de cabecera.

2.4.3 Interferencia de aire

Concierno a las comunicaciones entre la estación móvil y la BSS en los protocolos de las capas física, MAC, y RLC.

Las subcapas RLC/MAC permiten una eficiente multiplexación multiusuario en los canales de paquetes de datos compartidos, y utiliza un protocolo ARQ selectivo para transmisiones seguras a través del interfaz aire. El canal físico dedicado para tráfico en modo paquete se llama PDCH(Packet Data Channel).

En adelante se considerará la capa de enlace de datos (Data Link Layer) y la capa física (Physical Layer) como parte del Interface Aire Um.

2.4.4 Capa de enlace de datos

Capa de enlace de datos. Se encuentra entre la estación móvil (el móvil GPRS en sí) y la red.

Dividida en:

- La capa LLC (entre MS-SGSN): Provee un enlace altamente fiable, está basado en el protocolo DIC e incluye control de secuencia, entrega en orden, control de flujo, detección de errores de transmisión y retransmisión. Es básicamente una adaptación del protocolo LAPD usado en GSM.
- La capa RLC/MAC (entre MS-BSS): Incluye dos funciones. El principal propósito de la capa de Control de Radio Enlace (RLC) es la de establecer un enlace fiable. Esto incluye la segmentación y reensamblado de las tramas LLC en bloques de datos RLC y ARQ (peticiones de retransmisión) de códigos incorregibles. La capa MAC controla los intentos de acceder de un MS a un canal de radio compartido por varios MS. Emplea algoritmos de resolución de contenciones, multiplexación de multiusuarios y prioridades según la QoS contratada.

2.4.5 Capa física

Capa física entre MS y BSS. También se subdivide en dos subcapas.

- La capa del enlace físico (PLL) provee un canal físico. Sus tareas incluyen la codificación del canal (detección de errores de transmisión, corrección adelantada (FEC), indicación de códigos incorregibles), interleaving y la detección de congestión del enlace físico.

- La capa de enlace de radio frecuencia (RFL) trabaja por debajo de la PLL e incluye la modulación y la demodulación.

2.5 Packet Data Protocol (PDP)

Es una estructura de datos presentes tanto en el Nodo de Servicio de Apoyo GPRS (SGSN) y el nodo de pasarela GPRS de apoyo (GGSN) que contiene información del abonado sesión cuando el suscriptor tiene una sesión activa. Cuando un móvil quiere usar GPRS, primero debe conectar y activar un contexto PDP. Esto asigna un contexto PDP estructura de datos en el SGSN que el suscriptor se encuentra de visita y el GGSN sirve punto de acceso del suscriptor. Los datos registrados incluyen

- Suscriptor dirección IP
- Abonado IMSI
- Suscriptor
- Punto final del túnel ID (TEID) en el GGSN
- Punto final del túnel ID (TEID) en el SGSN
-

El ID de punto final del túnel (TEID) es un número asignado por el GSN, que identifica los datos relacionados con el túnel de un contexto PDP particular. [7]

2.6 Acces Point Name (APN)

Es un protocolo informático que permite por lo general el equipo del usuario para acceder a Internet utilizando la red de telefonía móvil.

A nivel técnico se trata de un identificador de red configurable utilizado por un dispositivo móvil para conectarse a un GSM operador. El transportista se examinará este identificador para determinar qué tipo de conexión de red debe ser creado, por ejemplo: ¿qué dirección IP debe ser asignado a un dispositivo inalámbrico, lo que los métodos de seguridad se debe utilizar, y cómo / o si, debe estar conectado a alguna red de clientes privados.[8]

Más concretamente, el nombre de punto de acceso (APN) identifica una dirección IP de paquetes de red de datos (PDN), que un usuario móvil de datos quiere comunicar. Además de la identificación de un PDN, una APN también se puede utilizar para definir el tipo de servicio (por ejemplo, la conexión a Wireless Application Protocol (WAP), servicio de mensajería multimedia (MMS)), que es proporcionada por el PDN. El APN se utiliza en 3GPP redes de acceso de datos, por ejemplo, General Packet Radio Service (GPRS), Evolved Packet Core (EPC). [8].

2.7 Point to Point Protocol

En las redes, el protocolo punto a punto, o PPP, es un protocolo de enlace de datos de uso común en el establecimiento de una conexión directa entre dos nodos de red. Puede proporcionar autenticación de la conexión, la transmisión de privacidad cifrada y la compresión.[9]

PPP se utiliza en muchos tipos de redes físicas con cable serie incluido, línea telefónica, la línea troncal, teléfono celular, enlaces especializados de radio, y enlaces de fibra óptica, tales como SONET. [9]

PPP también se utiliza en conexiones de acceso a Internet (en la actualidad se comercializa como "banda ancha"). La mayoría de los proveedores de servicios Internet (ISP) se utiliza PPP para el cliente en el acceso telefónico a Internet. Dos formas de encapsulado PPP, Point-to-Point Protocol over Ethernet (PPPoE) y Point-to-Point Protocol sobre ATM (PPPoA), se utilizan más comúnmente por los proveedores de servicios Internet (ISP) para establecer una línea de suscriptor digital (DSL) conexión a Internet de servicios con los clientes.[9]

PPP se utiliza comúnmente como un protocolo de capa de enlace de datos para la conexión a través de circuitos sincrónicos y asincrónicos, donde se ha sustituido en gran medida de la edad del Protocolo Internet de línea serie (SLIP) y la compañía telefónica normas de cumplimiento obligatorio (como el Protocolo de

acceso al enlace, equilibrado(LAPB) en la protocolo X.25 suite). PPP fué diseñado para trabajar con múltiples protocolos de capa de red, incluyendo “Internet Protocol” (IP), “Internetwork Novell Packet Exchange” (IPX), NBF y AppleTalk [9].

2.8 Socket

En las redes de computadoras, un Internet socket o network socket es un extremo de una asociación bidireccional entre procesos de flujo de la comunicación a través de un Protocolo de Internet basado en red informática , tales como el Internet .[10]

El término sockets de Internet también se utiliza como un nombre para una interfaz de programación de aplicaciones (API) para el TCP / IP stack de protocolo, por lo general proporcionada por el sistema operativo. Sockets de Internet constituyen un mecanismo para la entrega de paquetes de datos a la aplicación adecuada proceso o hilo, basado en una combinación de locales y remotos direcciones IP y números de puerto. Cada toma se asigna por el sistema operativo a un proceso de solicitud de comunicación o hilo. [10]

Una dirección de socket es la combinación de una dirección IP (la ubicación de la computadora) y un puerto (que se asigna al proceso de programación de aplicaciones) en una sola identidad, al igual que uno de los extremos de una conexión telefónica es la combinación de un número de teléfono y una extensión en particular [10].

2.9 Firewall

Es un dispositivo o conjunto de dispositivos diseñados para permitir o denegar las transmisiones de red basado en un conjunto de reglas y se utiliza con frecuencia para proteger las redes del acceso no autorizado al tiempo que permite las comunicaciones legítimas de pasar [11].

Muchos de computadoras personales los sistemas operativos incluyen software basado en servidores de seguridad para proteger contra las amenazas de la Internet pública. Muchos routers que pasar los datos entre las redes contienen

componentes de firewall y, a la inversa, muchos firewalls pueden realizar las funciones básicas de enrutamiento [11].

2.10 Protocolo de Transferencia de Archivos (FTP)

Es un protocolo de red estándar que se utiliza para transferir archivos de un host a otro en una red basada en TCP, como el Internet. FTP está construido sobre una arquitectura cliente-servidor y utiliza el control por separado y las conexiones de datos entre el cliente y el servidor. Los usuarios de FTP pueden autenticarse con un inicio de sesión en texto del protocolo, pero puede conectarse de forma anónima si el servidor está configurado para permitirlo [12].

Las primeras aplicaciones de cliente de FTP son interactivas herramientas de línea de comandos, la aplicación de los comandos estándar y la sintaxis. Clientes de interfaz gráfica de usuario han sido desarrollados para muchos de los sistemas operativos de escritorio más populares en la actualidad [12].

2.11 Protocolo de Control de Transmisión (TCP)

Es uno de los principales protocolos de la suite de protocolo de Internet. TCP es uno de los dos componentes originales de la serie, complementando el Protocolo Internet (IP), y por lo tanto, todo el conjunto se conoce comúnmente como TCP / IP. TCP ofrece una entrega confiable, ordenada de un flujo de bytes de un programa en un ordenador a otro programa en otro ordenador [13].

TCP es el protocolo que las principales aplicaciones de Internet como el World Wide Web, correo electrónico, administración remota y transferencia de archivos confiar. Otras aplicaciones que no requieren de un servicio fiable de flujo

de datos, puede usar el User Datagram Protocol (UDP), que proporciona un datagrama servicio que hace hincapié en reducir la latencia de más de fiabilidad [13].

2.12 Protocolo de Internet (IP)

Es una etiqueta numérica asignada a cada dispositivo (por ejemplo, ordenador, impresora) que participan en una red informática que utiliza el protocolo de Internet para la comunicación. Una dirección IP tiene dos funciones principales: host o una red interfaz de identificación y ubicación frente. Su papel se ha caracterizado de la siguiente manera: " A nombre de... indica lo que buscamos una dirección donde se indica una ruta indica cómo llegar hasta allí" [14].

Los diseñadores del protocolo de Internet definen una dirección IP como de 32-bit número, y este sistema, conocido como Protocolo de Internet versión 4 (IPv4), está todavía en uso hoy en día. Sin embargo, debido al enorme crecimiento de la Internet y la predicción de agotamiento de las direcciones disponibles , un nuevo sistema de direccionamiento (IPv6), de 128 bits para la dirección, fue desarrollado en 1995, estandarizados como RFC 2460 en 1998, y se está implementando en todo el mundo desde mediados de la década de 2000 [14].

Las direcciones IP son números binarios, pero normalmente se almacenan en archivos de texto y se muestra en legible anotaciones, como 172.16.254.1 (para IPv4), y 2001: db8: 0:1234:0:567:8:1 (por IPv6) [14].

La Internet Assigned Numbers Authority (IANA) gestiona la asignación de direcciones IP a nivel mundial del espacio y de los delegados cinco registros regionales de Internet (RIR) para asignar bloques de direcciones IP a los registros locales de Internet (Internet Service Provider) y otras entidades [15].

2.13 HTTP Cliente y Servidor

Describe la relación de los programas de cooperación en la aplicación. El componente de servidor dispone de una función o servicio a uno o varios clientes, que inician solicitudes de tales servicios [16].

Con funciones tales como el intercambio de correo electrónico, acceso a Internet y acceso a bases de datos, se basan en el modelo cliente-servidor. Usuarios que acceden a los servicios bancarios desde su ordenador utiliza un cliente de navegador web para enviar una solicitud a un servidor web en un banco. Ese programa puede a su vez remitirá la solicitud a su propio programa de cliente de base de datos que envía una petición a un servidor de base de datos en otro equipo del banco para recuperar la información de la cuenta. El saldo se devuelve al cliente de base de datos del banco, que a su vez sirve de nuevo al cliente del explorador Web que muestra los resultados al usuario. El modelo cliente-servidor se ha convertido en una de las ideas centrales de la computación en red. Muchas aplicaciones de negocios que se escriben hoy en día utilizan el modelo cliente-servidor. Lo mismo ocurre con los protocolos principales de Internet de aplicaciones, tales como HTTP, SMTP, Telnet y DNS [17].

La interacción entre el cliente y el servidor se describe a menudo usando diagramas de secuencia. Los diagramas de secuencia se han estandarizado en el Lenguaje de Modelado Unificado[16].

Los tipos específicos de clientes se incluyen navegadores web, clientes de correo electrónico y chat en línea los clientes. Los tipos específicos de servidores incluyen los servidores web, servidores FTP, servidores de aplicaciones, servidores de bases de datos, servidores de nombres, servidores de correo, servidores de archivos, servidores de impresión y servidores de terminales [16].

2.14 Global System Mobile (GSM)

Es un conjunto estándar desarrollado por el European Telecommunications Standards Institute (ETSI) para describir las tecnologías de segunda generación (o "2G") digital de las redes celulares. Desarrollado como un reemplazo de la primera generación de redes celulares analógicas, el estándar GSM descrito originalmente una cámara digital, conmutación de circuitos de la red optimizada para full duplex de voz de telefonía. La norma se amplió el tiempo para incluir circuito enciende por primera vez el transporte de datos, de transporte de paquetes de datos a través de GPRS. Velocidades de transmisión de paquetes de datos se incrementaron más tarde a través de EDGE. El estándar GSM es seguido por la tercera generación (o "3G") UMTS estándar desarrollado por el 3GPP. Redes GSM se desarrollará aún más a medida que comienzan a incorporar la cuarta generación (o "4G") LTE Advanced normas. "GSM" es una marca comercial propiedad de la Asociación GSM [18].

La GSM Association estima que las tecnologías definidas en la norma GSM servir el 80% de la población mundial, que abarca más de 5 millones de personas en más de 212 países y territorios, siendo GSM la más ubicua de las numerosas normas para redes celulares [18].

2.15 Global Position System (GPS)

Es un espacio basado en sistema mundial de navegación por satélite (GNSS) que permite la ubicación de la información y el tiempo en todo tiempo y lugar en o cerca de la Tierra, donde hay una línea de visión sin obstáculos a cuatro o más satélites GPS. Es mantenida por la de los Estados Unidos el gobierno y es de libre acceso por cualquier persona con un receptor GPS con algunas limitaciones técnicas que sólo se eliminan para los usuarios militares [19].

El proyecto de GPS fue desarrollado en 1973 para superar las limitaciones de los anteriores sistemas de navegación, la integración de las ideas de varios

predecesores, incluyendo una serie de clasificar los estudios de ingeniería de diseño de la década de 1960. GPS fue creado y realizado por el Departamento de Defensa de EE.UU. (USDOD) y se ejecutó inicialmente con 24 satélites. Que entró en pleno funcionamiento en 1994 [19].

Además de los GPS, los sistemas de otros están en uso o en desarrollo. El ruso Sistema Global de Navegación por Satélite (GLONASS) se encontraba en su uso por el ejército ruso, hasta que fue totalmente a disposición de los civiles en 2007. También hay prevista la china Brújula sistema de navegación y de la Unión Europea el sistema Galileo de posicionamiento [19].

2.16 Invernaderos

Es un edificio donde se cultivan las plantas. Estas estructuras varían en tamaño desde pequeños cobertizos para edificios muy grandes. Un invernadero en miniatura que se conoce como un marco frío [20].

Un invernadero es una estructura con diferentes tipos de materiales de revestimiento, tales como un vaso o en el techo de plástico y con frecuencia de vidrio o paredes de plástico, que se calienta por la radiación solar visible de entrada (para los que el vidrio es transparente) del sol es absorbida por las plantas, del suelo, y otras cosas en el interior del edificio. Aire calentado por el calor de las superficies calientes interior se conserva en el edificio por el techo y la pared. Además, las estructuras calienta y las plantas dentro del invernadero re-irradian algo de su energía térmica en el infrarrojo, a la que el vidrio es opaco, en parte, por lo que parte de esta energía es atrapada en el interior del invernadero. Sin embargo, este último proceso es un actor menor en comparación con el primero (convectivas) del proceso. Así, el mecanismo principal de calefacción de un invernadero es la convección. Esto se puede demostrar mediante la apertura de una pequeña ventana cerca del techo de un invernadero: la temperatura baja considerablemente. Este principio es la base de

la Autovent sistema de enfriamiento automático. Por lo tanto, el cristal usado para un invernadero trabaja como una barrera al flujo de aire, y su efecto es atrapar energía dentro del invernadero. El aire que se calienta cerca del suelo, se evita el aumento de forma indefinida y que fluye de distancia [20].

2.17 Microcontroladores

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de entrada y salida, es decir, se trata de un computador completo en un solo circuito integrado, estos son diseñados para disminuir el costo económico y el consumo de energía de un sistema en particular, (Palacios 2004). Por eso el tamaño de la CPU, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. A modo de ejemplo el control de un electrodoméstico sencillo como una licuadora, utilizará un procesador muy pequeño de 4 bits (un byte es un conjunto de 8 bits, un megabyte son 1024 bytes). En cambio un reproductor de música digital mp3 requerirá de un procesador de 32 bit, el control de un sistema de frenos ABS (Antilock Brake System) se basa normalmente en un microcontrolador de 16 bit, al igual que el sistema de control electrónico del motor en un automóvil (Valencia 2009).

Los modernos microcontroladores, frecuentemente incluyen un lenguaje de programación integrado, como el Basic con plataforma de desarrollo PIC Basic Pro Compiler v2.6 que se utiliza bastante con este propósito o el PCW en C. Los microcontroladores negocian la velocidad y la flexibilidad para facilitar su uso. Debido a que se utiliza bastante sitio en el chip para incluir funcionalidad, como los dispositivos de entrada y salida o la memoria que incluye el microcontrolador, se ha de prescindir de cualquier otra circuitería, (Bodington 2008).

2.18 Software

El software es el soporte lógico de una computadora, son las instrucciones básicas de control de hardware controladores o drivers (parte física o tangible de una computadora) y que puede estar desarrollado en alto nivel, entendimiento con el usuario, o parte visual. Un sistema operativo es el programa intermediario entre el controlador de hardware y el programa visual utilizado por el usuario.

Los sistemas operativos de la empresa desarrolladora de software Microsoft han aparecido desde el primer computador IBM 8088 en la década de los 80's de manera comercial para uso domestico con su famoso MS-DOS, evolucionando posteriormente al famoso Windows y sus variedades siendo una de las más famosas de los últimos años la xp, dicha versión cuenta con un respaldo impresionante de desarrollo en software de toda índole, pero con mayor auge la de comunicación con los puertos de la computadora para utilizarla como control de todo tipo de procesos industriales de automatización (Ceballos 2007).

Durante los últimos 20 años el lenguaje de programación C ha sido el elegido para desarrollar aplicaciones comerciales para control y de negocios, este lenguaje proporciona un alto grado de control al programador permitiéndole el uso de funciones de bajo nivel otorgando poder y flexibilidad. El Framework .NET proporciona un conjunto de herramientas que ayudan a elaborar código que funcione con la herramienta .NET desarrollada por Microsoft y C++ es uno de los lenguajes evolucionados y actualizados del famoso C, de modo tal que C++ cuenta con todo el poder de C, Microsoft ha escrito desde cero respetando su sintaxis, orientado a objetos y uso de clases, eliminando los apuntadores y con uso de funciones encadenadas al núcleo del sistema operativo Windows de Microsoft (Ferguson *et al.*, 2003). Matlab fue un software originalmente escrito para desarrollar matrices fácilmente en 1970, actualmente Matlab es un lenguaje de alto rendimiento técnico de computo que proveen herramientas para realizar algoritmos matemáticos complejos, modelado, simulación, graficas de ingeniería, análisis de datos exploración y visualización, procesamiento de imágenes, entre otras. (Sivanandam *et al.*, 2007).

III. MATERIALES Y MÉTODOS

En base a la investigación realizada y documentación acerca de la tecnología GPRS (General Packet Radio Services), se crea una base de administración de datos para los invernaderos automatizados, esta base es generada a partir de las necesidades que se tengan dentro del mismo invernadero.

Esta base de datos serán administrados a través de un microcontrolador quien a base de una programación seriada identificará que acción tiene que realizar dentro del mismo, para tener siempre estable el invernadero, pero estas acciones se reciben a través de otra plataforma de comunicación, la cual es una tarjeta con modulo GPRS de la compañía TELIT.

Las ventajas que encontramos en esta tarjeta es la versatilidad que tiene en la administración de los datos pues cuenta con ocho canales de transmisión de este protocolo. Esta tarjeta de desarrollo es preprogramada por la misma compañía pero a un nivel alto pues solo el usuario necesita enviarle comandos para que las instrucciones ya establecidas respondan según lo que se la haya enviado.

3.1 Ubicación del área del proyecto

Universidad Autónoma de Querétaro. El pueblo de Amazcala pertenece al municipio del Marqués, Qro. el cual se localiza en el sector Suroeste del estado, entre los 20° 31' y 20° 58' de latitud Norte. Su longitud se encuentra entre los 100° 09' y los 100° 24' del Oeste a 1850 m sobre el nivel del mar. Colinda al Oeste con el municipio de Querétaro, al Norte con el estado de Guanajuato, al Este con el municipio de Colón y al Sur con los municipios de Huimilpan y Pedro Escobedo. Las carreteras disponibles para llegar el poblado de Amazcala son la carretera 57, la carretera a Chichimequillas y México libre. La temperatura media oscila entre

los 18 y los 24°C, con un clima predominante subtropical, templado semiseco (INEGI, 2005), Figura 3.1



Figura 3.1. Ubicación geográfica del proyecto.

3.2 Especificación de la instalación del equipo

Será colocado en un cuarto que está construido en plano con firme de tepetate, suelo en concreto con piso de bloque de azulejo, esta precedido por un cuarto sala con dimensiones de 3.30 m de ancho por 2.80 m de largo por 2.40 m de altura, tiene una ventana fija de 90 cm de ancho por 90 cm de alto y una puerta de 80 cm de ancho por 2.05 m e alto de metal. La habitación está construida con muros de tabique de concreto hueco con enjarre y pintura en blanco, el techo es de ladrillo hueco de 25 cm con cama de concreto e impermeabilizado, sus medidas son de 3.30 m de ancho por 3.30 m de largo por 2.40 m de alto, tiene una única puerta de metal de 80 cm de ancho por 2.05 m de alto con media ventana de cristal fija, cuenta con 8 tomas de corriente en 127 v, la puerta abre a la sala ya descrita y esta está orientada al norte geográfico.

3.3 Instalación y especificación del equipo

Dentro de la habitación ya mencionada se encuentra un equipo de cómputo, el cual se utiliza para que el usuario administre de forma directa los datos que requiera del invernadero, es decir será el control principal del equipo, el cual a través de una interfaz gráfica se pudo hacer que la comunicación fuese de forma sencilla para el operador. El equipo de cómputo se comunica a través del protocolo RS232 con una tarjeta electrónica que contiene el microcontrolador 18F4550 Microchip, el cual es el responsable directo de hacer las acciones requeridas que se piden por parte del usuario, a su vez existe otro microcontrolador que se conecta al equipo GM862-GPS TELIT.

El módulo GM682-GPS TELIT es una parte medular de la administración de los datos pues a través de este se da aviso de cualquier acción que sea requerida por parte del usuario en donde este se encuentre. Los datos van a ser transferidos a través de paquetes de radio a un móvil o a una página de internet donde se pueda observar dicha información.

Es importante mencionar que el equipo además de contar con el módulo GPRS, provee un servicio de mensajería a través de SMS (Short Message Service) siendo indispensable para dar alertas al usuario que contenga poca información, además el equipo tiene un servicio de GPS (Global Position System) tiene aplicaciones directas a la posición en la que se encuentra nuestro equipo y se le ha dado distintas aplicaciones no solo al estar en un invernadero de proporciones muy grandes sino que además esta parte del desarrollo de la administración de datos fue utilizada para dar ubicación a móviles de construcción como lo son los Tractores a los cuales se les envían datos vía GPRS de las coordenadas geográficas en las cuales estos deben realizar las labores a las cuales fueron asignados, reduciendo de esta manera el costo que implicaba que el operador fuera a realizar un trabajo no planeado en otra parte no asignada.

3.4 Descripción del módulo GM862-GPS TELIT

El módulo GM862-GPS TELIT cuenta con las siguientes características:

3.4.1 Comunicación con el módulo GM862-GPS TELIT

El equipo GM862-GPS TELIT puede ser controlado a través de una interface serial usando el estándar de los comando AT. La familia del modulo inalámbrico Telit está en conjunto con los siguientes documentos:

1. Conjunto de comandos AT, para mantener la compatibilidad con programas existentes SW.
2. 3GPP TS 27.007 especifica los comandos AT y comandos específicos para GPRS.
3. 3GPP TS 27.005 especifica los comandos AT para SMS(Short Message Service) y CBS (Cell Broadcast Service).
4. FAX Class 1 Compatibilidad de comandos.

Las reglas de la sintaxis para la implementación de Telit de otros módulos que contengan comandos AT y comandos FAX son muy similares al estándar básico. Un comando especial es #SELINT, que ha sido introducido para tener una interface AT muy cercana al estándar uno.

Existen dos tipos de extensión de comandos:

Comandos de parámetros: Este tipo de comandos quizá deban ser de “escritura” (para almacenar un valor o valores de uso posterior), “lectura” (determina el valor actual o valores almacenados), o “prueba” (se utiliza para determinar los rangos de los valores de apoyo). Cada uno de ellos tiene un comando de prueba (trailing

= ?) que da información acerca del tipo de este subparámetro; además tienen un comando de lectura (trailing ?) para verificar los valores de subparámetros.

Comandos de acción: Este tipo de comandos quizá deban de ser de “ejecución” o “prueba”. “Ejecución” para invocar una función en particular del equipo, el cual generalmente involucre más que un simple almacenamiento de un valor para uso posterior. “Prueba” determina el estado del subparámetro actual de prueba.

Para poder comunicarnos con el módulo se requiere una información estructurada pues la comunicación es serial. Una línea de comandos se compone de tres elementos: el prefijo, el cuerpo y la terminación de carácter.

El prefijo consta de los caracteres “AT” o “at”.

El cuerpo es aquel subparámetro al que se quiere hacer referencia ejemplo “AT\$GPSACP”, nos da información de la ubicación actual de la antena GPS.

La terminación de carácter, con la cual se indica que se ha terminado de escribir el parámetro y queremos obtener la información deseada; por default está definido <CR> (carring return).

Para que el usuario pueda obtener la información deseada, el modulo se encuentra en comunicación constante con un microcotrolador PIC 18F4550 Microship, el cual ha sido programado con los comando adecuados para que le estructura sea enviada de manera correcta como lo requiere modulo, y a su vez la respuesta que nos el GM862-GPS debe ser interpretada por el microcontrolador para obtener información completa.

Para poder iniciar comunicación por primera vez con el modem Telit, es necesario conectarlo a la computadora a través del puerto serial, y utilizar una interfaz que nos permita hacer las configuraciones correctas para este caso se ha utilizado el software “HyperTerminal”. Figura 1.



Figura 1. Hyperterminal

A través de esta herramienta nosotros podemos hacer las configuraciones necesarias para que el modem responda a las requerimientos que nosotros tenemos en distintas aplicaciones.

Primero le damos un nombre a nuestra sesión y le asignamos un icono. Figura 2.



Figura 2. Asignación de un nombre e icono.

Seguido de esto nos aparecerá una segunda ventana para la configuración definir el tipo de comunicación que se hará a través de la computadora. Figura 3.

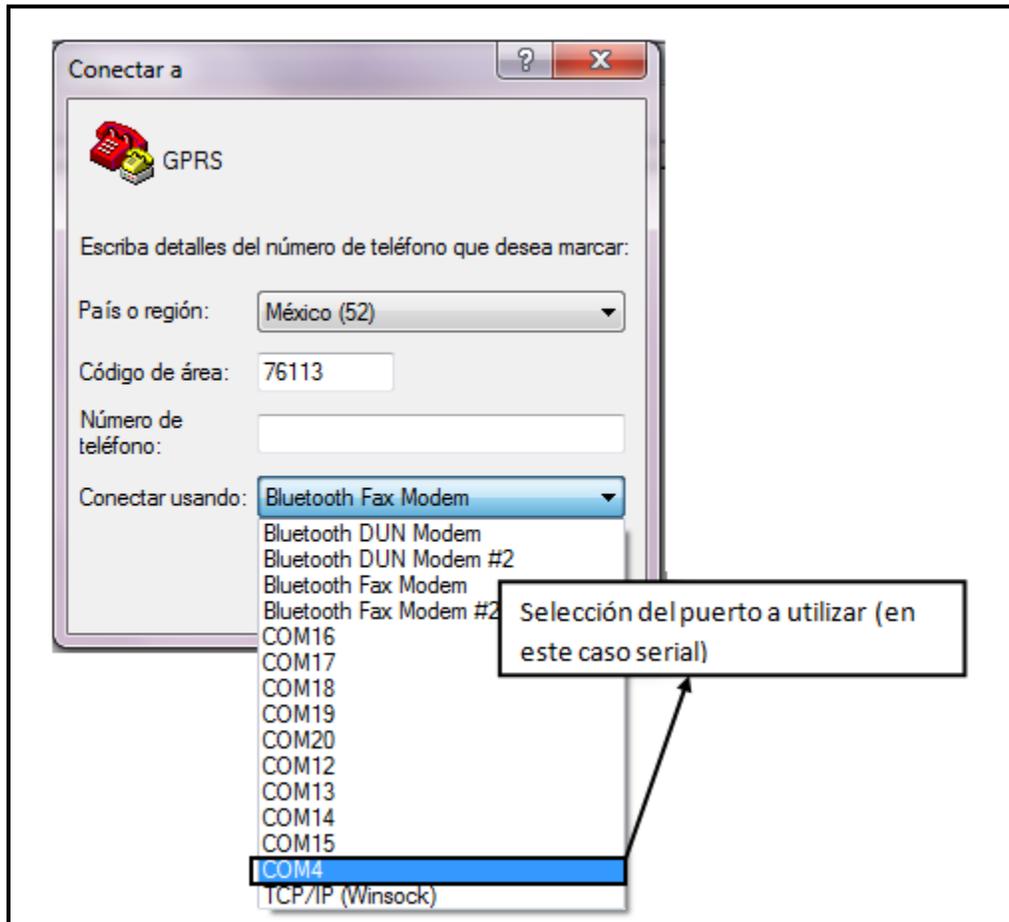


Figura 3. Selección del puerto a utilizar.

Posterior a la selección del puerto de comunicación aparecerá una ventana para configurar los parámetros y las características de comunicación que se desee tener por parte del usuario. Figura 4.

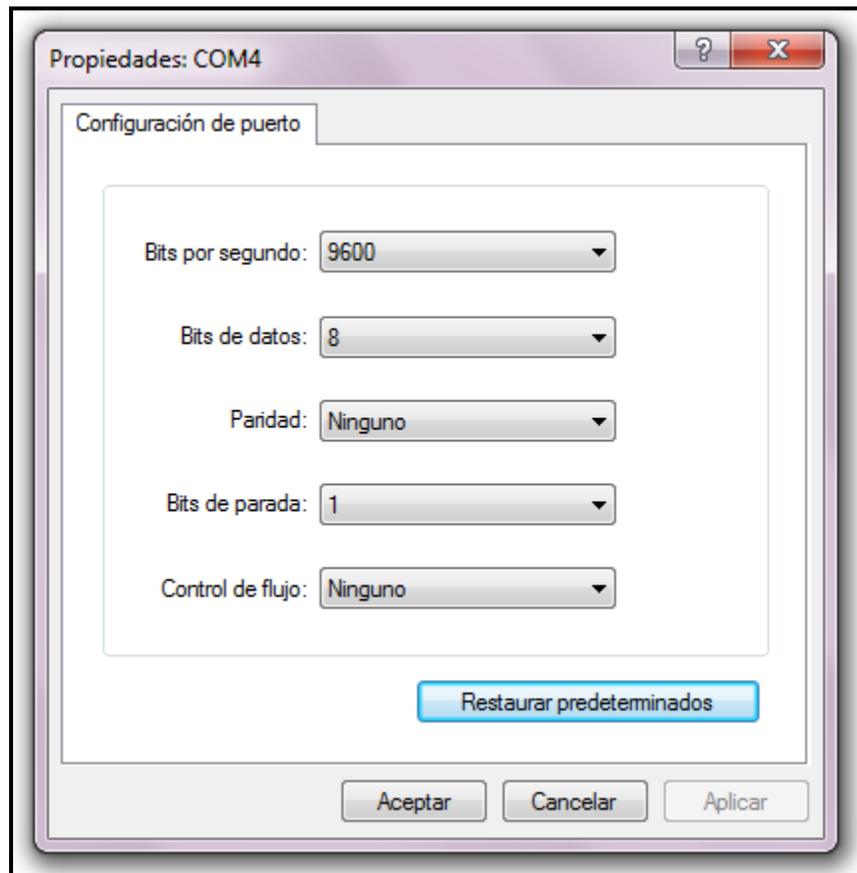


Figura 4. Configuración de las características de transmisión serial.

Una vez que se haya concluido esta parte es además necesario configurar la interpretación de la información que será enviada a través del puerto así como la recepción de la misma. Para ello dentro del programa HyperTerminal se puede acudir a: Archivo->Propiedades aparecerá una ventana en donde se tendrá que mover a la opción de configuración y dentro de ella dar click al botón "Configuración ASCII" para hacer los correspondientes cambios de acuerdo a lo necesario. Figura 5.

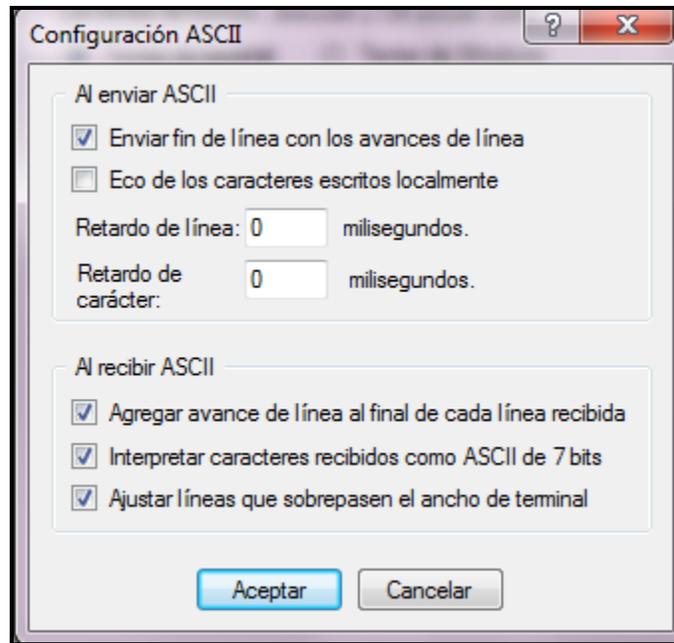


Figura 5. Configuración de interpretación de datos.

Una vez realizadas todas las configuraciones y características deseadas es posible interactuar con el modem, cabe señalar que es indispensable hacer esta comunicación primero antes de pasar al desarrollo de los sistemas, pues a partir de la información que ofrecen los comandos y la manera o forma de interpretar las respuestas dadas será la base en la manera de hacer los programas más adelante pues cada uno de los sistemas necesita distinta programación de acuerdo a las funciones que serán requeridas por parte de cada uno de ellos.

Se probará con el comando AT, donde el modem da una respuesta de OK (Figura 3.6), esta respuesta es dada por la interpretación que se le dio a la transmisión de datos, pero si se observa la tabla 1, la transmisión de datos interpretada es a través de código hexadecimal y es posible ver que al momento de ejecutar el enter o <Carriage Return> la respuesta inmediata por parte del modem es enviar un <Carriage return>+<Linefeed character> (<CR>+<LF>)y posteriormente la palabra OK seguida de un <CR>+<LF>, es importante el conocimiento de la transferencia de datos pues en ocasiones puede ser el inicio de una cadena de errores desfavorables cuando se requiere interpretar la información de manera automática

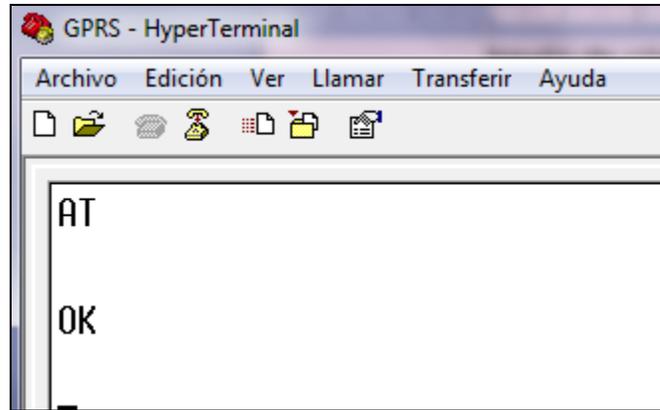


Figura 6. Respuesta en ASCII del comando solicitado.

HEX	ASCII
0D 0A	<CR><LF>
0D 0A	<CR><LF>
4F 4B	OK
0D 0A	<CR><LF>

Tabla 1. Comparación de respuestas en código Hexadecimal y ASCII

En el caso exclusivo de la respuesta OK por parte del modem tenemos dos <CR><LF> anteriores a la palabra ya mencionada y uno posterior.

Nota: es posible ver el código hexadecimal a través de SLOW la cual es una aplicación que viene dentro del software PIC C Compiler. Figura 7.



Figura 7. Aplicación PIC C Compiler

3.4.2 Comandos AT necesarios a utilizar:

Para poder realizar un control de invernaderos adecuado se requiere del conocimiento de ciertos códigos AT para posteriormente utilizarlos en los softwares que tendrán esta aplicación. Estos comandos se han dividido de acuerdo a su función; es decir se han agrupado aquellos que estén relacionados de manera general con el módulo TELIT GM862-GPS, algunos otros se dividen de acuerdo a la señal específica que utilizan: GSM, GPRS y GPS.

Aplicaciones Generales:

AT: Se espera la respuesta OK si el modulo está funcionando de manera adecuada.

AT#SELINT: Se utiliza para seleccionar la configuración de modelo TELIT, con este comando muchas aplicaciones se configuran en automático.

AT&Z: Utilizado para guardar números en la SIM del teléfono.

AT+CGMI: Proporciona la identificación de la marca utilizada.

AT+CGMM: Se identifica el modelo del dispositivo utilizado.

AT+CGMR: Número único para identificar el dispositivo.

AT+CGSN: Revisión del dispositivo.

AT+CIMI: Identificación internacional subcrita del móvil.

Aplicaciones GSM:

AT+COPN: Lectura de todos los operadores de red.

AT+CREG: Proporciona el estado de registro de la red.

AT+CLKC: Notifica la hora configurada o se puede configurar a la hora deseada.

AT+CFUN: Comando necesario para poder definir la transmisión de datos por el puerto serial del módulo TELIT.

AT+CSQ: Como resultado de la ejecución de este comando se obtiene el nivel de señal.

AT+CMER: Comando para activar o desactivar el envío de resultados de códigos para la asistencia técnica.

AT+CALA: Es el comando para configurar una alarma, está se configura de acuerdo al tiempo que se requiera dar algún aviso y se puede hacer por periodos repetitivos de manera automática o solo una vez.

AT+CBC: Este comando ejecuta un resultado del nivel actual de la carga de la batería.

AT+CMGF: A través de este comando se selecciona el formato de escritura del mensaje de texto.

AT+CNMI: Indicación de un nuevo mensaje existe.

AT+CMGL: Regresa un reporte del estatus que se tenga de mensajes almacenados.

AT+CMGS: Es el comando correcto para poder enviar un mensaje de texto, según los parámetros que hayan sido configurados anteriormente.

AT+CMGD: Con este comando es posible eliminar algún mensaje que se tenga en memoria.

AT#GSMCONTCFG: Comando utilizado para configurar el envío de mensajes a través de la red GPRS.

AT#SMSATRUN: Habilita o deshabilita el servicio de SMS.

Aplicaciones GPRS:

AT+CGCLASS: Selección del tipo de clase para la red GPRS.

AT+CGATT: Configuración para adjuntar o separar el servicio GPRS.

AT+CGEREP: Este comando se utiliza para obtener un reporte extendido de algún posible error arrojado del servicio GPRS.

AT+CGREG: Con este comando es posible registrar el modulo al servicio de la red que este detecte.

AT+CGQMIN: Permite establecer los parámetros mínimos necesarios para que el funcionamiento del modulo sea adecuado cuando se requiera conectar este a la red.

AT+CGACT: Con este comando se habilita el contexto PDP (Packet Data Protocol), que posteriormente será el primer llamado para una identificación rápida en la apertura de un socket.

AT+CGPADDR: Regresa la dirección IP asignada una vez que se haya configurado el APN de la red se la cual se solicitará el servicio.

AT+CGDATA: Si es necesario utilizar el servicio PPP, este es el comando indicado pues relaciona este servicio con el contexto definido PDP.

AT#AUTOATT: Se configura de manera automática el adjuntar o separar el servicio GPRS

AT#BND: Seleccionar de manera manual la banda deseada para la transmisión de datos.

AT#AUTOBND: Se configura de manera automática la banda con la cual el modulo estará enviando o recibiendo datos.

AT#GAUTH: Se configure a través de este comando el tipo de autenticación a utilizar.

AT#TCPATRUNCFG: Comando importante cuando se habla de la transmisión de datos a través de GPRS, pues con este mismo se puede configurar los elementos necesarios para obtener una buena conexión.

AT#SS: Reporta el estado actual que tienen los sockets de conexión.

AT#SI: Se ejecuta un reporte con los datos de los sockets, es decir notifica cuales han sido el total de bytes descargados a través del mismo

AT#SGACT: Se utiliza para activar el contexto PDP

AT#SH: Es posible apagar el socket con este comando.

AT#SCFG: Se ejecutan los parámetros de configuración del socket.

AT#SRECV: Utilizado para recibir datos de la red en modo comando.

AT#SSEND: Comando utilizado para enviar datos por el socket activado a la red, y específicamente al puerto de conexión actual.

AT#SGACTCFG: Se relaciona la configuración emitida para el socket con los parámetros de contexto activado por el PDP.

AT#USERID: Se configure un nombre para el usuario.

AT#PASSW: Se configure la contraseña para el usuario.

AT#PKTSZ: Se configura el tamaño de paquete para la recepción de los datos.

AT#GPRS: Activa el contexto GPRS, si ya se ha utilizado el comando #SGACT, en forma automática se realizará la activación del servicio de paquetes de datos.

AT#FTPOPEN: Utilizado para abrir el puerto del cual se extraerá el archivo deseado.

AT#FTPCLOSE: Con este comando es posible cerrar el puerto de conexión.

AT#FTPPUT: Se puede subir un archivo a la red.

AT#FTPGET: Se consigue algún archivo.

AT#FTPTYPE: Se selecciona el modo de transmisión.

AT#FTPGETPKT: Ejecución de comandos, emitida durante una conexión FTP en donde se abre una conexión de datos y se comienza a recibir un archivo desde el servidor FTP mientras el mismo permanece en modo de comando.

AT#FTPREST: Se configura a un parámetro de fábrica en donde es posible cancelar la descarga de algún archivo en cuestión.

Aplicaciones GPS

AT\$GPSACP: Comando mayormente utilizado para obtener la posición global, además envía la hora actual (este parámetro corresponde únicamente a la hora del meridiano de Greenwich), es posible obtener la cantidad de satélites con los cuales la antena de GPS pudo hacer contacto para obtener dicha información.

3.4.3 Descripción de la tarjeta electrónica de comunicación

Esta tarjeta ha sido diseñada para este proyecto, en ella se incorpora un microcontrolador PIC 18f4550 Microship, su sistema de alimentación con un cristal de 4 mhz, conexión para el puerto serie RS-232 a la computadora, su alimentación es de 5 v continuos al PIC y por medio del mismo puerto de 12 v continuos, un botón de reinicio físico, sus ocho salidas A y B del PIC en forma de pines sencillos dobles y sus entradas analógicas y digitales, figura 8.

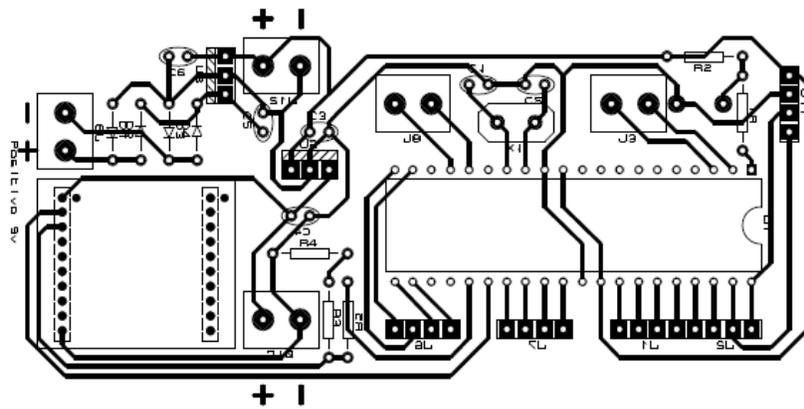


Figura 8. Diagrama electrónico del esclavo.

Esta tarjeta es donde se localiza el maestro y que está conectada además de la computadora al modem, figura 3.10, esto es posible a través de un bus de datos realizado a partir de una compuerta lógica AND 7408, que a partir de una serie de comandos emitidos por el microcontrolador, es posible referenciar quien será el destinatario de la información enviada.

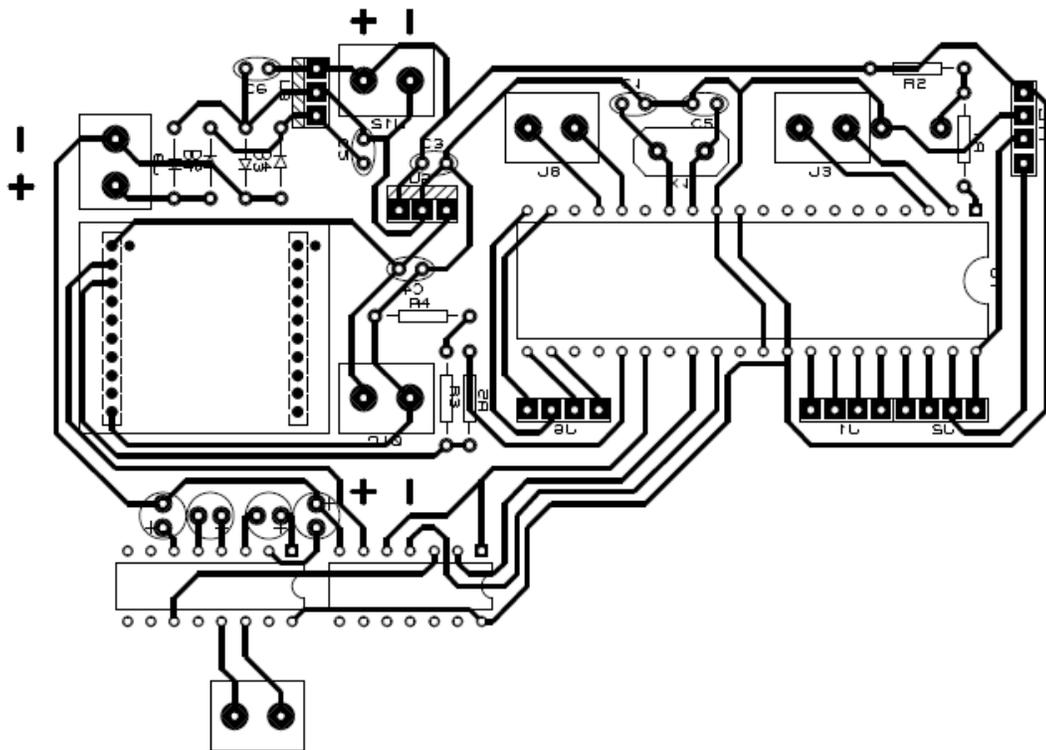


Figura 10. Diagrama electrónica de la placa maestra

La función principal de esta tarjeta es contener y dar un óptimo funcionamiento al microcontrolador el cual como ya se ha mencionado fue seleccionado por las características que lo integran, este funciona como el maestro en la operación, pidiendo y enviando datos que hayan sido requeridos para su uso posterior.

3.4.4 Descripción de la tarjeta electrónica GM862-GPS TELIT

Para que el módulo GM862-GPS pueda funcionar de manera adecuada se requiere una tarjeta electrónica capaz de dar esos requerimientos, por las características del módulo de transmisión de información, la tarjeta electrónica cuenta con un apropiado suministro de energía a través de un zócalo de conexión que da una entrada de 18 volts pero es llevado a 12 volts por un regulador electrónico, contiene además salidas para conectar este módulo con la tarjeta SIM (una característica del mismo es que el mismo módulo tiene entrada para dicha tarjeta), existe una conexión UART directa, tiene salidas para colocar audífonos y estar en manos libres, se colocó 8 entradas y salidas de Propósito General por las características del módulo. Figura 11.



Figura 11. Modem GM862-GPS

Provee una solución completa para aplicaciones de datos y teléfonos. La tarjeta electrónica presenta una serie de soluciones, la cual cubre los requisitos existentes del módulo.

El modulo Telit contiene una tarjeta electrónica base en donde es posible colocar el modem de dicha marca para realizar las operaciones correspondientes, además de incluir una antena GPS, y otra antena para GSM/GPRS.

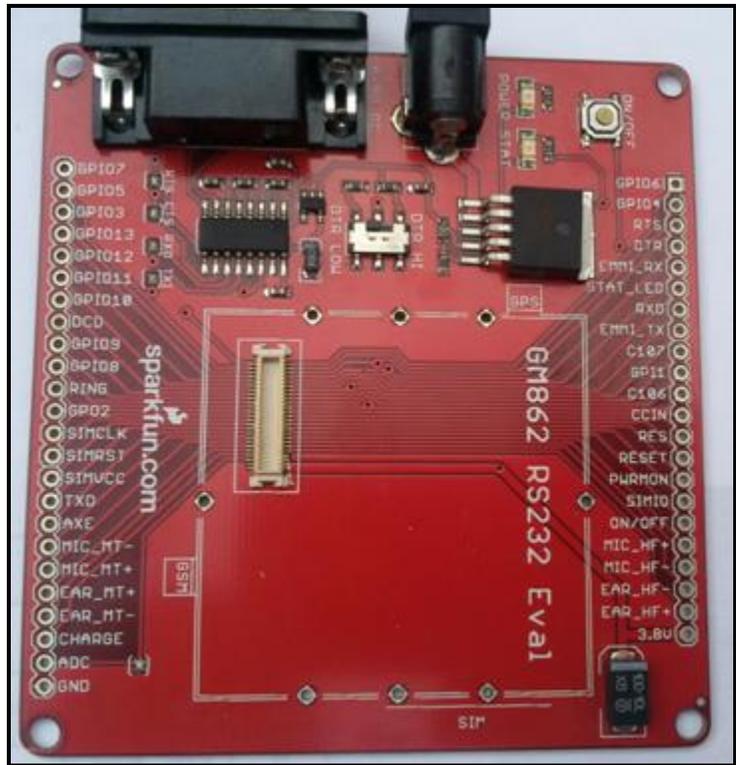


Figura 12. Tarjeta electrónica para el modem GM862-GPS



Figura 13. Antena GPS

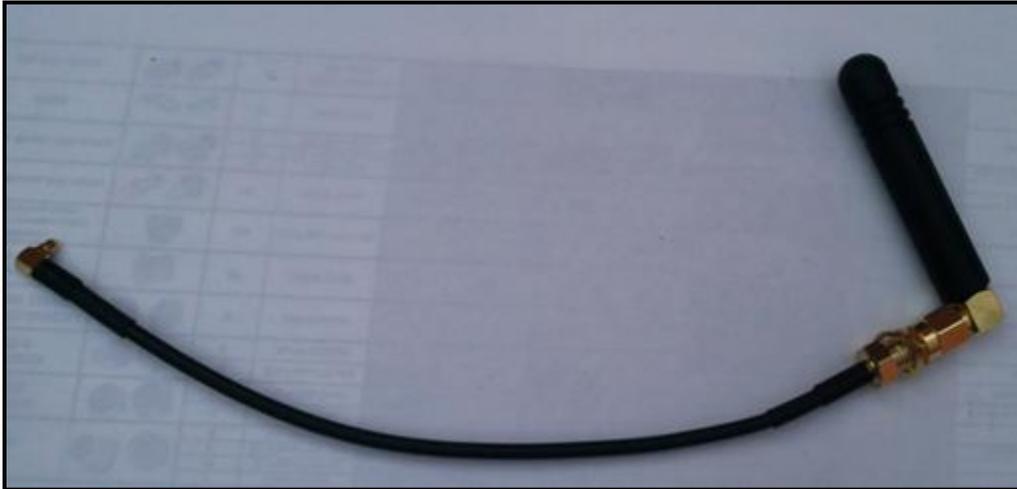


Figura 14. Antena GSM

3.5 Desarrollo del software de para la administración de datos

Se desarrollaron un conjunto de software necesarios para la administración de los datos; han sido divididos de tres maneras: la aplicación directa en una interfaz de computador, el programa que contiene el microcontrolador para poder ser el maestro y principal ejecutor de la operación, y por último se encuentra la interfaz gráfica en el celular, necesario para un usuario en movimiento.

Cada uno de los programas integra de manera específica el producto pues se requiere la participación de ellos para tener un buen control sobre el invernadero.

3.5.1 Programación del Microcontrolador

El software del microcontrolador ha sido creado a través del compilador PIC C Compiler versión 4.088 en lenguaje C. El microcontrolador 18f4550 (Figura 15) una vez colocado en la tarjeta electrónica de comunicación correcta, realiza

actividades indispensables para la operación, pues tiene la capacidad para recibir una instrucción y realizar una acción a ella. Se comunica directamente con la computadora a través del puerto serial, en donde se dan las instrucciones necesarias hacia el microcontrolador conectado directamente a la computadora en la habitación de control.

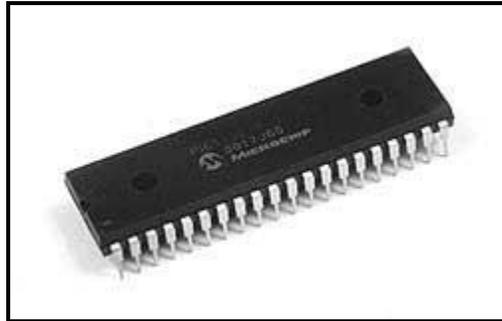


Figura 15. PIC 18f4550

El programa para el microcontrolador maestro se muestra a continuación, y puede ser resumido en la conexión directa con el modulo GM862-GPS y una antena ZigBee para poder enviar información hacia sus esclavos.

La programación física del microcontrolador solo es posible con una interface que permita el grabar un código en el microcontrolador para ello se ha utilizado el Master Prog. Figura 16.

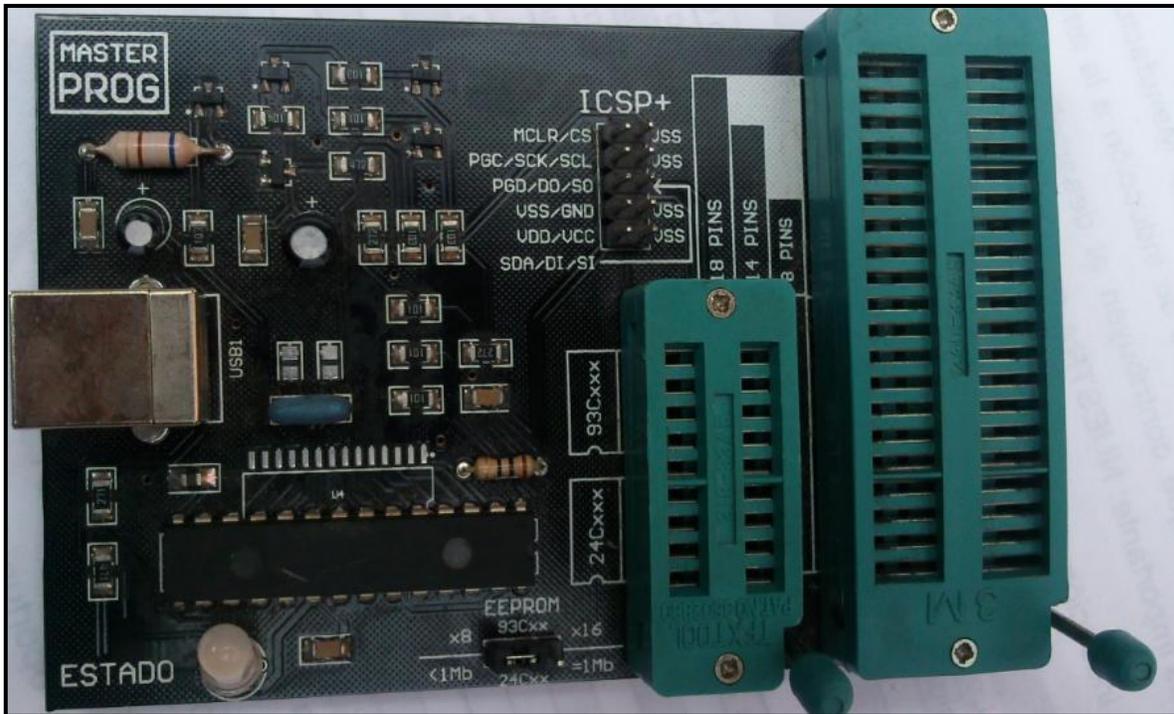


Figura 16. Master Prog, utilizado para el grabado de código en el PIC.

Programa Maestro.

En la programación en C siempre es necesario colocar las librerías para que el programa responda conforme a lo que se requiere.

```
#include <18F4550.h>
#include <string.h>
#include <stdlib.h>
#fuses XT,NOPROTECT,NOWDT
#use delay(clock=4000000)
#use rs232(rcv=PIN_C7,xmit=PIN_C6,baud=9600,bits=8,parity=N)
#define LCD_DATA_PORT getenv("SFR:PORTD")
#include <lcd.c>
#define buf_length 128
```

Par el efecto de este programa se ha declarado algunas variables globales pues el repetido uso de ellas hace necesaria esta aplicación

```

int
buf[buf_length],str[buf_length],done[buf_length],go[buf_length],x[
buf_length],y[buf_length];
int i=0,str_flag=0, t=0, s=0,j=0, n=0, g=0, b=0, c=0, f=0,l=0,m=0;
int a=0,e=0,coma=0,h=0,k=0,bn=0,cn=0,activar=0;
float norest=0,west=0;
char platica;

```

Se ha declarado algunas variables utilizadas por el modem para a partir de la identificación del variable en cuestión el programa pueda hacer una ejecución determinada.

```

int atg[9]={'$', 'G', 'P', 'S', 'A', 'C', 'P', ':', 0};
int gps[11]={'A', 'T', '$', 'G', 'P', 'S', 'A', 'C', 'P', 0x0D, 0x0A};
int
smse[22]={'A', 'T', '+', 'C', 'M', 'G', 'S', '=', '"', '4', '4', '2', '1', '4',
'3', '0', '3', '8', '8', '"', 0x0D, 0x0A};
int smsr[2]={'>', 0};
int at[5]={'A', 'T', 0x0D, 0x0A, 0};
int ok[4]={'O', 'K', 0x0D, 0};
int sigue[3]={1, 0, 0};
int ctrz[1]={0x1A};
int comillas[1]={'"'};

```

Se han declarado funciones para poder analizar las tramas de dato por separado e incluso vaciar las variables que sirven como buffers.

```

void descargar();
void analisis();
void enviar_sms();
void enviar_gps();

```

Interrupción que se ejecuta cuando una nueva trama de datos es enviada a través del puerto serial.

```

#int_rda
void rcv_data()
{
    buf[i]=getchar();

    if(buf[i] == 0x0A)
    {
        i=0;
        t++;
    }
}

```

```

else if(buf[i] == 0x0D)
{
    s++;

    if(s == 3)
    {
        a++;
        strcpy(str,buf);
        f=1;
        i=0;
    }

    for(j=0; j<= 127; j++)
    {
        buf[j]=0;
    }
    i=0;
    j=0;
}

else if(buf[i]== '>')
{
    bn=1;
    for(j=0; j<= 127; j++)
    {
        buf[j]=0;
    }
    i=0;
    j=0;
    buf[0]='>';
    buf[1]=0;
    strcpy(go,buf);
    buf[0]=0;
    buf[1]=0;
}

else
    i++;
}

```

La interrupción serial es una función importante pues a partir de la constante interrupción de la misma será posible el reconocimiento adecuado de los caracteres que se estén enviando en el momento justo.

Otras función que contiene el programa, por su aplicación es fundamental para el funcionamiento del mismo; esta función es el análisis, pues a partir de esta función se reconoce que tipo de variable se va estar manipulando y con ello que acción inmediata se va a tomar por parte del controlador.

```

void analisis()
{
    int u;

    f=0;
    for(u=0;u<=127;u++)
    {
        done[u]=str[u];
        if(done[u]!=':')
        {
            e++;
            done[u+1]=0;
            strcpy(go,done);
        }
    }
}

```

Para no tener problemas con la comunicación, especialmente cuando se almacenan las variables en buffers es altamente recomendable que estos mismos sean vaciados por que en ocasiones llegan a guardar datos impidiendo que la siguiente información sea fiable , a continuación se agrega dicho función:

```

void descargar()
{
    for(c=0;c<=126;c++)
    {
        str[c]=0;
        buf[c]=0;
        done[c]=0;
        go[c]=0;
        x[c]=0;
        y[c]=0;
    }

    s=0;
    i=0;
    h=0;
    k=0;
    coma=0;
    l=0;
    m=0;
    //activar=0;
}

```

La operación que toma su importancia por mandar la instrucción al módulo TELIT es la de conexión a GPRS, donde es necesario colocar el APN correcto de nuestro proveedor, pues puede provocar errores si no se es conocido bien este dato.

Además del ingresar de manera correcta el APN, es necesario activar el contexto PDP, el cual es una identificación por la cual se puede configurar características

para su conexión a internet, a continuación se agrega el código que hace posible esta acción:

```
void conectar ()
{
    printf("AT+CDCONT=");
    printf("internet.itelcel.com");

    printf(SIM900,"webgprs");

    printf(SIM900,"webgprs2003");
    imprimir(0);
    checar=strcmp(dato,ok);
    descargar();
    lcd_gotoxy(20,4);
    lcd_putc('4');
}
```

3.5.2 Programación del software PC

El programa que fue diseñado para ser instalado en una computadora, ha sido creado a través de la interface Visual C# de Microsoft 2010 (Figura 17), Se selecciono este lenguaje de programación dada su gran flexibilidad y eficiencia, este es de distribución gratuita desarrollado por Microsoft y con licencia libre, uno de los puntos por el cual fue utilizado es su importancia en la manera que podrá ser transportado de ser requerido a otras plataformas, modificado y actualizado. De momento el software cumple con los requisitos para el proyecto, que es el manejo de datos vía GPRS.



Figura 17. C#

La programación es orientada a objetos, lo cual hace posible que tenga una interfaz gráfica son mejores características que en otras plataformas. Figura 18.

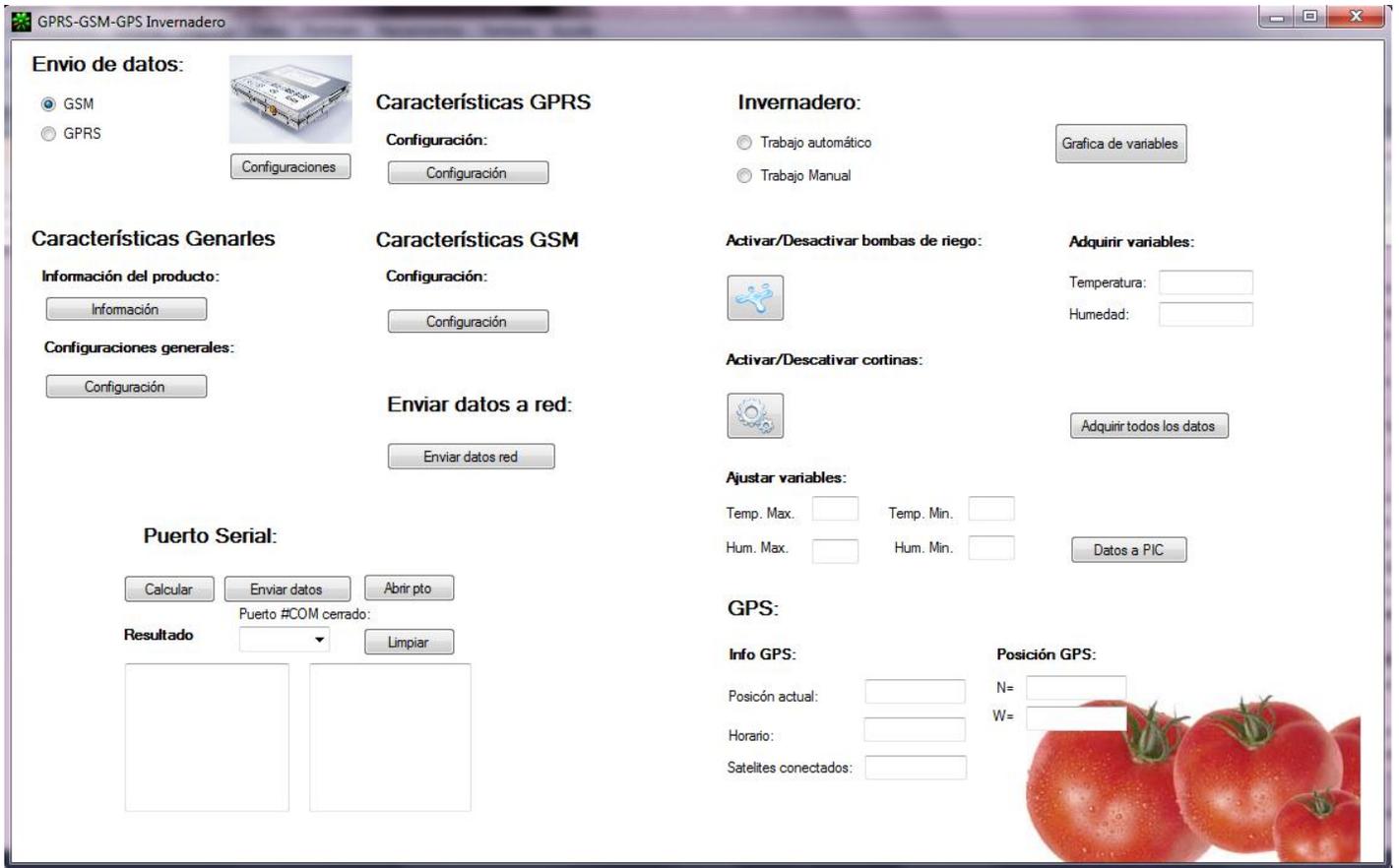


Figura 18. Ventana principal de la administración de datos.

Esta es la interfaz que el usuario estará utilizando para poder pedir los datos que se necesiten para un control adecuado del invernadero. Cuenta con algunas parámetros de configuración pues de esta manera el usuario podrá definir bien cualquier contexto de conexión que requiera, además de poder enviar y recibir datos según las aplicaciones del mismo.

El programa se ha creado de acuerdo a las diferentes aplicaciones de la red y de la configuración que se tenga por parte del módulo de conexión TELIT. Se puede configurar tres distintas herramientas necesarias para la aplicación del control y la información del dispositivo

Se encuentra un área de características generales (Figura 19.) por medio de las cuales se puede obtener información importante que será necesaria revisar para poder trabajar de manera adecuada con el programa mismo y con el módulo de conexión

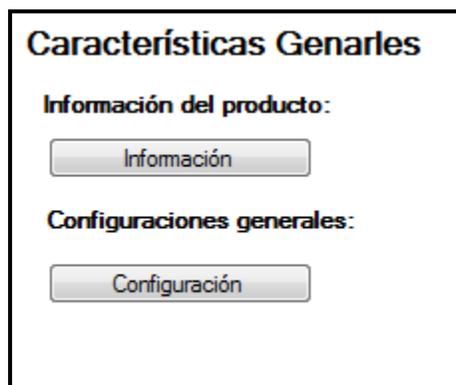


Figura 19. Parámetros de configuración

Si se le da click al botón información, se abrirá una pantalla con los siguientes datos, ver figura 20.

The image shows a software window titled "Infogene" with a standard Windows-style title bar. The main content area is titled "Información básica del producto:". Below this title, there are several input fields, each with a label to its left:

- SIM Insertada en Modem: [input field]
- Calidad de la señal: [input field]
- Carga de batería: [input field]
- Identificación de marca: [input field]
- Identificación de revisión: [input field]
- Identificación de movil internacional: [input field]
- Estado de actividad: [input field]
- Fecha: [input field]
- Hora: [input field]

At the bottom right of the form area, there is a button labeled "Conseguir Datos".

Figura 20. Conseguir información.

Uno de los puntos importantes cuando se adquiere la información es el detectar si la tarjeta SIM esta insertada de manera correcta, y si lo está una respuesta de READY será obtenida pero de acuerdo al procesamiento que hace en primer instancia el microcontrolador solo será enviada una respuesta de "insertada" o "no insertada", cabe mencionar nuevamente que el maestro es microcontrolador pues enviará notificaciones cuando sea conveniente según la ya programado.

Estos parámetros de información solo se pueden conseguir cuando se haya seleccionado un puerto serial con las configuraciones correctas. Ver figura 21.

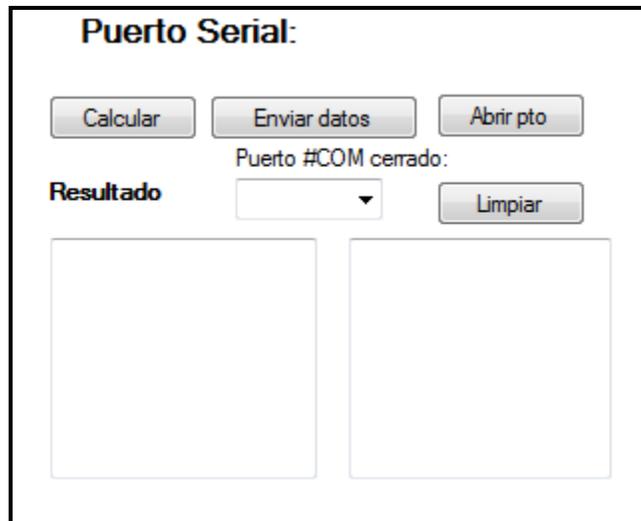


Figura 21. Conexión serial.

Una vez realizada una correcta selección de puerto serial, según el cual esté conectado al hardware maestro es decir donde se encuentra conectado el microcontrolador, el módulo TELIT y la antena zigbee.

Cuando se ha dado click en el botón conseguir datos (figura 20), se manda la información en forma de cadena por el puerto serial de todos los datos que se ven en la ventana de información, se envía una cadena de datos a la vez ya que cada uno de estos parámetros requiere un comando AT distinto y además es necesario poner un periodo de tiempo pues en algunas ocasiones es posible perder información importante para poner una respuesta adecuada.

El siguiente código explica cómo se conecta de manera serial y como se piden los datos:

```
//Se abre el puerto seleccionado
private void button4_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)//Si está abierto
    {
        serialPort1.Close();//Cerrar puerto
    }
}
```

```

        //Actualizar botón
        btnabrir.Text = "Abrir Puerto";

        //Actualizar etiqueta
        label12.Text = "Puerto COM " + cbpuertos.SelectedItem.ToString() +
"cerrado";
    }
    else//Si está cerrado
    {
        //Primero configuramos las propiedades del puerto
        serialPort1.PortName = cbpuertos.SelectedItem.ToString();

        //Ahora podemos abrirlo
        serialPort1.Open();

        //Actualizar btnabrir
        btnabrir.Text = "Cerrar Puerto";
    }
}

```

Cuando se da click en el botón de conseguir datos de la figura 21 se envía una serie de código al microcontrolador para que este a su vez pueda enviarlos al módulo:

//Al momento de conseguir los datos se colocan de manera automática en una caja de texto y se pone entre cada dato enviado un retraso para el siguiente envío de comando, es retraso se hace con el fin de la llegada completa de los datos, y la probable perdida de los mismo pues se estaría ejecutando de manera inmediata la interrupción en el microcontrolador incluso antes de poder conseguir información para algún parámetro en específico.

```

private void btninfo_Click(object sender, EventArgs e)
{
    Infogene info = new Infogene();
    DialogResult resulta = info.ShowDialog();
    if (resulta == DialogResult.OK)
    {
        txtr.AppendText("AT+CPIN\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CSQ\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CBC\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CGMI\r\n");
        System.Threading.Thread.Sleep(100);
    }
}

```

```

        txtr.AppendText("AT+CGMM\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CGMR\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CGSN\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CIMI\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CPAS\r\n");
        System.Threading.Thread.Sleep(100);
        txtr.AppendText("AT+CCLK\r\n");
        System.Threading.Thread.Sleep(100);
    }
    info.Dispose();
    info = null;
}

```

//Los datos son primero colocados en una caja de texto para posteriormente enviarlos.

```

private void button3_Click(object sender, EventArgs e)
{
    serialPort1.WriteLine(txtr.Text);
}

```

//Al momento de recibir datos por parte del microntrolador son agregados a una caja de texto donde posteriormente se hace un análisis para enviar la información a cada área dependiendo de la aplicación pedida

```

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    //Da tiempo para que lleguen todos los datos al puerto
    System.Threading.Thread.Sleep(100);

    //Leer datos
    String datos = serialPort1.ReadExisting();

    //Cambiar la propiedad Text de txtentrada de manera segura
    tbre.Invoke(new EventHandler(delegate
    {
        tbre.AppendText(datos + "");
    }));
}

```

Otra característica importante con la cual cuenta el programa es la configuración general del módulo, pues a partir de esta información obtendremos un rendimiento óptimo para que responda conforme se han hecho los programas.

Si se observa en la figura 22 se encuentran con características como Seleccionar red inalámbrica que a partir de este parámetro se selecciona la red para poder trabajar con TA (Terminal Adapter). Otro punto importante es la configuración de la velocidad a la cual el módulo va a transmitir los datos con el microcontrolador, y con el programa de computador computadora misma la mejor elección aquí es seleccionar el reconocimiento de la velocidad automática que es la detección de la velocidad a la cual se está trabajando, esto evitará muchos problemas de comunicación pues simplemente el módulo se adapta a la configuración ya estipulada para las demás herramientas.

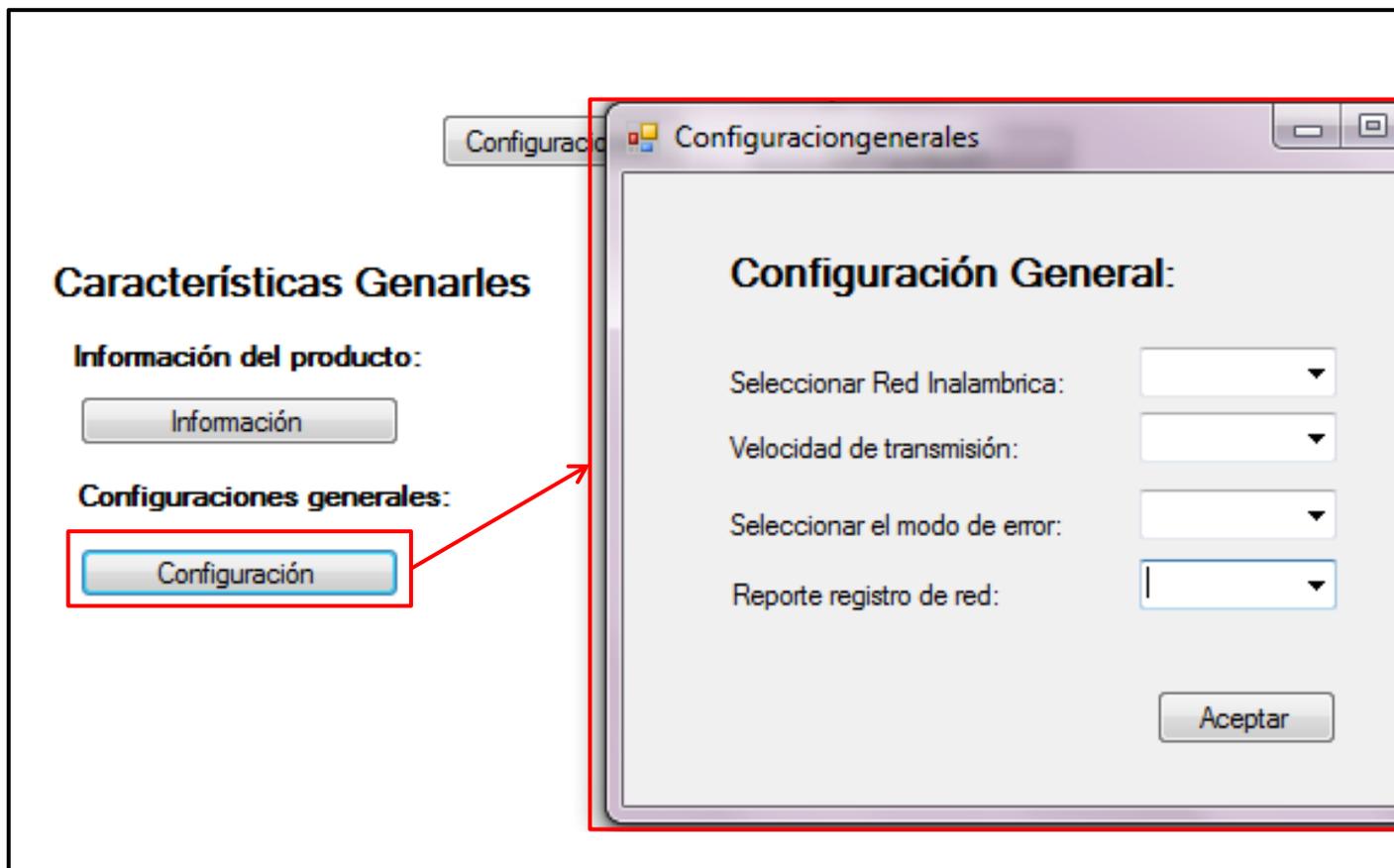


Figura 22. Configuración general

A continuación se dará la explicación a la parte principal del programa pues está cumple con el objetivo de comunicar un invernadero a través del servicio general de paquetes por radio (GPRS, General Packet Radio Service).

Existe en el programa un botón de configuración para tener en óptimas condiciones la comunicación. Ver figura 23.

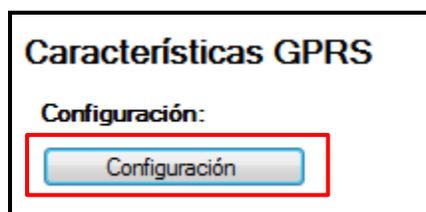


Figura 23. Configuración GPRS

Después de haber dado click sobre el botón de configuración como se observa en la figura anterior aparecerá la pantalla como se muestra en la figura 25, en la cual se ha dividido en tres partes principales que integran la configuración de la red GPRS, esto solo para poder poner los parámetros necesarios para conectarse a dicha red, los cuales van a cambiar dependiendo del proveedor al se tenga contratado, para el caso particular de esta investigación, se tiene un contrato con la empresa TELCEL (figura 24), estos datos se pueden fácilmente verificar haciendo uso del botón de información antes descrito y si es requerido hacerlo directamente solo basta con enviar el código AT+COPS? Que mostrará los datos de la red a la cual se conecta.



Figura 24. SIM Telcel.

Las áreas principales son configuración general, definición del contexto Packet Data Protocol, la calidad mínima de servicio así como la calidad requerida de servicio.

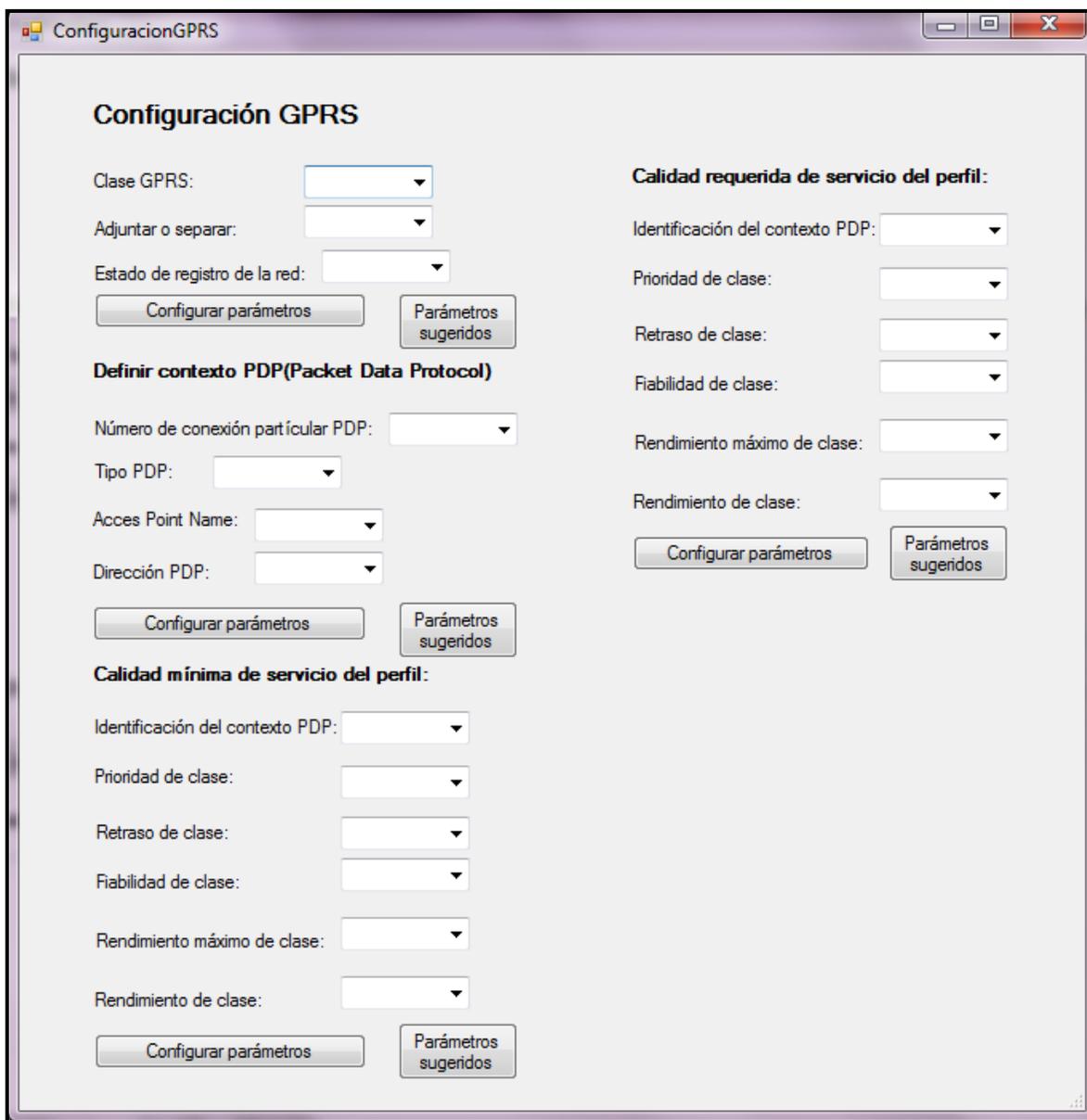


Figura 25. Configuración de los parámetros GPRS.

Para la configuración general (figura 26) es importante seleccionar la clase a utilizar, pues dependiendo de ella se podrán mandar los datos por la red GSM/GPRS, GPRS O solo GSM.



Figura 26. Configuración general

En la figura 26 se puede observar que para poder pedir los datos, es posible hacer la selección de algunos de ellos pues los “combobox” han sido programados para que se cuente con esa opción, pero si no se está seguro de que parámetros son los adecuados para que se tenga una buena transmisión de datos, se ha colocado un botón que contiene ya los parámetros predeterminados y ahorrara tiempo de investigación al usuario para que este se enfoque en otras actividades.

A continuación se presenta una parte del código donde se muestra como se realiza la configuración a parámetros predeterminados, y posteriormente como son llamados para ser enviados a través del puerto serial y recibir la indicación adecuada:

//Si se da click en el botón de parámetros sugeridos de manera automática son colocados los valores adecuados para el programa

```
private void button6_Click(object sender, EventArgs e)
{
    comboBox1.SelectedIndex = 0;
    comboBox2.SelectedIndex = 0;
    comboBox3.SelectedIndex = 1;
}
```

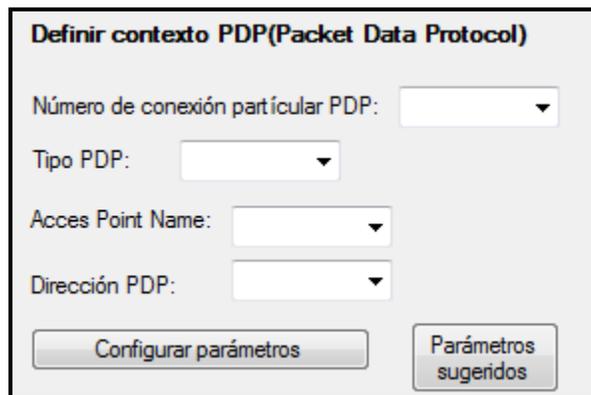
// Si la selección de estos parámetros se desea hacer de forma manual se enviará una relación de las características seleccionadas.

```
private void button1_Click(object sender, EventArgs e)
{
```

```
clasegprs = comboBox1.SelectedIndex;  
adjuntargprs = comboBox2.SelectedIndex;  
edored = comboBox3.SelectedIndex;  
}
```

La configuración más importante es la definición del PDP (Packet Data Protocol), ver figura 27, pues a partir de estas configuraciones, se podrá estar conectado a la red del proveedor, se necesita hablar con el mismo para que este pueda proporcionar la información de forma adecuada.

Para que el módulo pueda identificar una conexión y configurarla de acuerdo a lo requerido es necesario seleccionarlo, y se puede hacer desde 1 al 6, con la simple selección de números dentro de este rango es posible identificar y configurar.



The image shows a dialog box titled "Definir contexto PDP(Packet Data Protocol)". It contains four dropdown menus for configuration: "Número de conexión particular PDP:", "Tipo PDP:", "Acces Point Name:", and "Dirección PDP:". At the bottom, there are two buttons: "Configurar parámetros" and "Parámetros sugeridos".

Figura 27. Definición del contextp PDP

Otro punto muy importante será el tipo de transmisión que se realice pues, es posible hacerlo mediante el Internet Protocol (IP), el Acces Point Name (APN) es altamente recomendable preguntar al proveedor, pues si en este punto se cuenta con el APN correcto es posible que siempre se obtenga un resultado de error pues

la configuración no es la correcta, y no podremos utilizar la red contratada, para el caso particular del SIM que se utilizó los datos son los siguientes:

Compañía: TELCEL

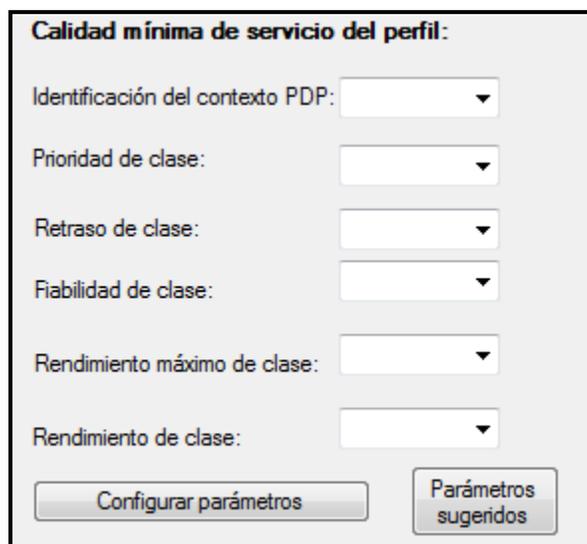
APN: internet.itelcel.com

Usuario: webgprs

Contraseña: webgprs2003

Algunas ocasiones es posible que se pida una contraseña así como un usuario, este dato se ha tomado en cuenta para la programación de los diversos software.

Pero para que se conecte además de tener un APN adecuado es necesario configurar al módulo con características indispensables para que la conexión se pueda realizar (figura 28).



Calidad mínima de servicio del perfil:

Identificación del contexto PDP:

Prioridad de clase:

Retraso de clase:

Fiabilidad de clase:

Rendimiento máximo de clase:

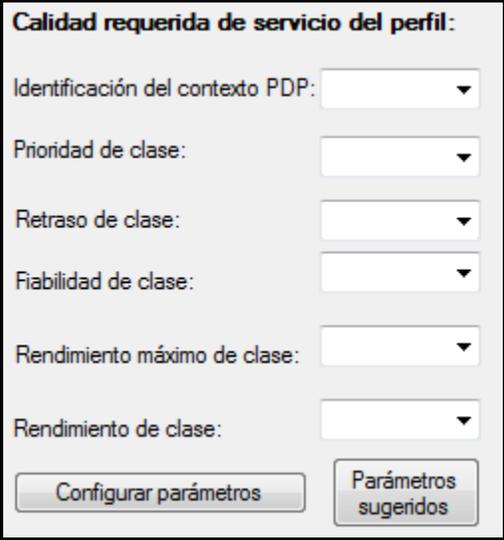
Rendimiento de clase:

Figura 28. Calidad mínima de servicio.

La calidad mínima de servicio para la comunicación, en donde es posible configurar características tales como prioridad de la clase, el retraso de la clase, la fiabilidad o el rendimiento todos estos parámetros deben estar dentro de un contexto permitido por lo cual para facilitar la aplicación al usuario se ha colocado

en cada condición diferentes opciones a elegir según sea la más adecuada, y si se desconoce del tema también se agrega un botón para ajustar a los parámetros predeterminados.

Al igual que la calidad mínima de servicio también es importante la calidad requerida del servicio donde se pueden configurar algunos pasos como la configuración anterior, pero la diferencia radica que esta es usada cuando se solicita el uso de la red para activar el contexto PDP, se envía en forma de mensaje.



Calidad requerida de servicio del perfil:

Identificación del contexto PDP:

Prioridad de clase:

Retraso de clase:

Fiabilidad de clase:

Rendimiento máximo de clase:

Rendimiento de clase:

Figura 29. Calidad Requerida

Para poder enviar datos a través de la red GPRS es activar el contexto PDP, así como abrir el socket necesario para poder hacer uso de la red.

En la figura 30 es posible observar las características necesarias a configurar entre ellas la activación de la identificación del socket, que asociará un contexto PDP previamente definido el cual se encuentra en un rango de 1 a 6, también importante configurar el tamaño del paquete que se va utilizar para intercambiar información vía TCP.

Configurar Socket:

Identificación de la conexión del socket:

Identificar contexto PDP (Packet Data Protocol):

Tamaño del paquete a utilizar por TCP: bytes

Máximo tiempo de intercambio (1-65535):

Máximo tiempo de espera de conexión (10 - 1200):

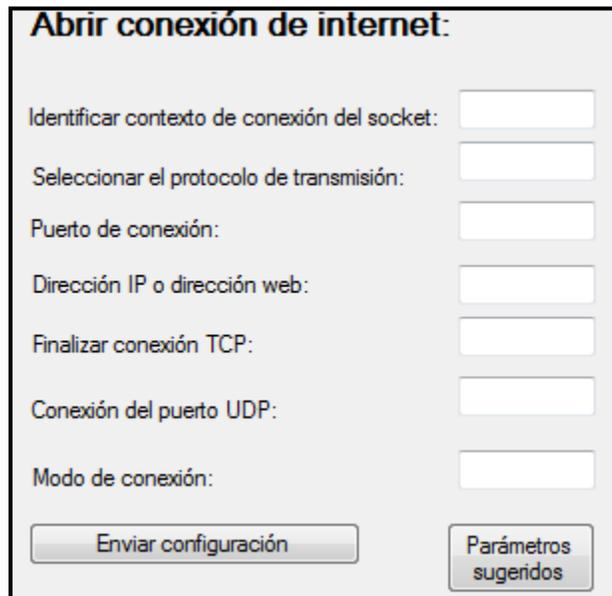
Tiempo de envío de datos (1 - 255):

Figura 30. Configurar socket.

Con el socket previamente configurado de manera correcta, ya es posible tener acceso a una página de internet, solo queda agregar unas cuantas líneas de comando más para poder descargar los datos de la misma.

Cuando se requiera conectar a internet se pedirá que sea a través de número de socket con los que cuenta el módulo que para el caso del TELIT GM862-GPS son 6, una vez identificado el socket, se pedirá además seleccionar el protocolo de transmisión el cual puede ser TCP o UDP, se tiene que elegir un puerto remoto de conexión este es un puerto con el que cuenta el módulo, requiere agregar una dirección IP o la dirección de la pagina web a la cual se desea ingresar, la forma

en que se quiera cerrar la conexión, elegir un puerto UDP, y seleccionar el modo de conexión el cual puede ser “modo online” o “modo comando”.



Abrir conexión de internet:

Identificar contexto de conexión del socket:

Seleccionar el protocolo de transmisión:

Puerto de conexión:

Dirección IP o dirección web:

Finalizar conexión TCP:

Conexión del puerto UDP:

Modo de conexión:

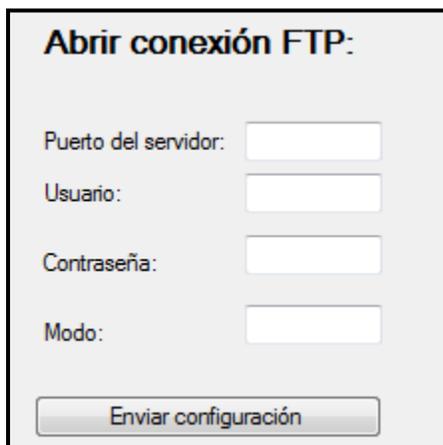
Figura 31. Abrir conexión de internet.

En este momento se mandarían los datos a través del puerto serial hacia el microcontrolador y este a su vez esperará una respuesta de “CONNECT” el cual significa que la conexión se ha hecho con éxito y se encuentra en ese momento navegando a través de internet.

Pero para poder hacer un control es necesario enviar datos, archivos, o descargarlos; para ello se ha creado en el programa una ficha interesante donde se puede abrir una conexión FTP con algún servidor para poder hacer la transferencia de archivos o la descarga de los mismo.

En la figura 32 se puede observar los parámetros a llenar; es necesario conocer el puerto del servidor, el usuario, la contraseña y el modo de transmisión.

Dependiendo de la programación del servidor es posible en este momento enviar algún archivo en formato de texto y procesarlo de forma adecuada.



Abrir conexión FTP:

Puerto del servidor:

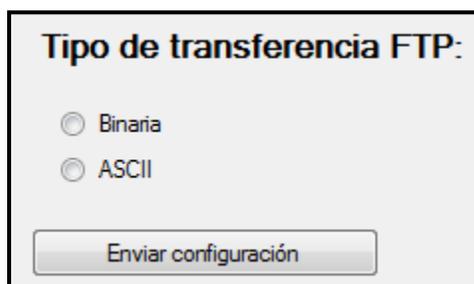
Usuario:

Contraseña:

Modo:

Figura 32. Abrir conexión FTP.

La transferencia de dicho archivo se puede hacer en dos formatos de manera binaria o en código ASCII.



Tipo de transferencia FTP:

Binaria

ASCII

Figura 33. Transferencia FTP.

Para las aplicaciones GSM, la configuración resulta sencilla en comparación del GPRS, pues solo es cuestión de configurar el servicio con el cual se va a trabajar la red GSM y los parámetros para que la comunicación sea posible se configuran de manera automática, ver figura 34.



Figura 34. Configuración GSM

Con respecto a la aplicación GPS también se puede configurar de forma más sencilla no se requiere más que el comando AT\$GPSACP para poder obtener la información relacionado con la posición global, la hora actual, la cantidad de satélites conectados y la altitud a la cual se encuentra localizada la antena de dicha señal.



Figura 35. Información GPS

Como se puede observar en la figura 35, solo es cuestión de obtener la información dándole click al botón "Posición Global", e inmediatamente sabremos la ubicación actual, así como la hora por parte del meridiano Cero o meridiano de Greenwich.

Por último en este programa se encuentra con sección para visualizar el comportamiento del invernadero como se ver en la figura 36:

Invernadero:

Trabajo automático Trabajo Manual

Grafica de variables

Activar/Desactivar bombas de riego:



Activar/Desactivar cortinas:



Adquirir variables:

Temperatura:

Humedad:

Adquirir todos los datos

Ajustar variables:

Temp. Max. Temp. Min.

Hum. Max. Hum. Min.

Datos a PIC

Figura 36. Información del invernadero.

Dentro de la información disponible para el invernadero se puede hacer uso de una aplicación automática o manual dependiendo del servicio que se requiera en ese instante, cuando se requiere mandar datos a un servidor a partir de la información actual recibida solo se agrupan todos los datos de manera serial para que puedan ser comunicados con el microcontrolador, este los procese los comunique con el módulo y a su vez este último lo envía a una servidor.

El programa es solo una interfaz actúa de acuerdo a las peticiones que realice el PIC maestro pues la comunicación entre ellos es serial, y el programa solo envía o recibe respuestas.

3.5.3 Programación del software para un móvil

El programa fue diseñado bajo la plataforma de Java en la versión Microedition es una herramienta que está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura

El programa muestra una interfaz gráfica fácil de utilizar por el usuario y contiene las herramientas necesarias para el control y manejo de los invernaderos.

Con las aplicaciones de java es posible hacer este tipo de programas pues dicha plataforma puede ser instalada en cualquier sistema operativo; para propósitos de programación primero se simulo sobre un teléfono celular virtual que viene integrado en el ambiente de programación de java me.



Figura 37. Programa en simulación.

El celular virtual tiene la forma como se muestra figura 37, en donde se puede simular la acción requerida, pero es importante señalar que no es posible verificar la conexión de la red a través de solo la simulación, como ya se señaló fue

simplemente utilizado para efectos de programación, es decir visualizar como sería programada la interfaz.

El programa para el celular (observar figura 37) solo se ha integrado la parte más crítica del control sobre el invernadero, como lo es el activar o desactivar las bombas, cortinas o alarmas.



Figura 38. Selección de host

Otra parte importante es la conexión que tendrá en la red existen distintos tipos de host programados según el requerimiento del usuario y a diferencia con el programa o la interfaz para una computadora personal (figura 38), en el celular hay características ya predefinidas, las mismas obtenidas directamente de las características del celular y de la red que está dada por el proveedor.



Figura 39. Seleccionar variable a monitorear

Una característica importante para el control (figura 39), es la selección de las variables a monitorear; el objetivo de la investigación es que estas variables sean adaptadas al tipo de aplicación que se requiera, es decir no es exclusivo del monitoreo solo de la temperatura y humedad, que es el requerido para un invernadero donde se practica la horticultura, puede ser diseñado para la acuicultura o cualquier otro tipo de producción primaria.

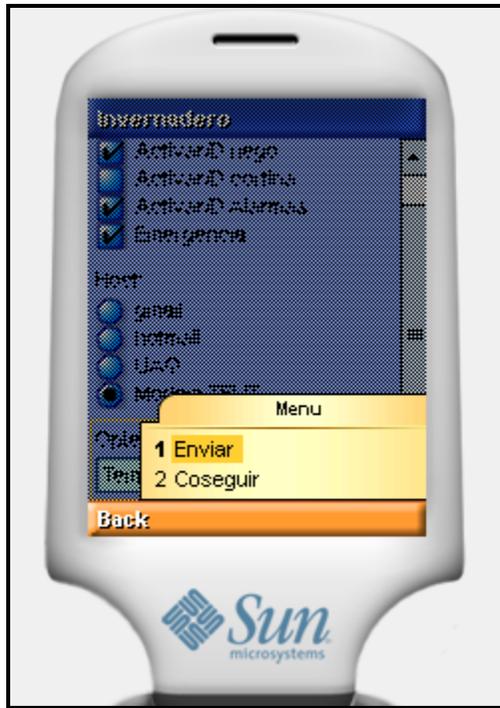


Figura 40. Acción requerida

Para poder decidir si queremos controlar o monitorear se ha instalado al programa dos clases de opciones en la cual se pueden pedir dichos datos o se pueden enviar, para una mejor apreciación visual ver figura 40.

IV. RESULTADOS Y DISCUSIÓN

4.1 Software

Resultado ser de fácil uso para el usuario dada su interfaz intuitiva, rápida ejecución de código y explicativo al usuario en el manejo de los posibles errores de operación detectados por el mismo.

Cada uno de los softwares creados para distintas aplicaciones han funcionado correctamente, en el caso del microcontrolador maestro realiza todas las acciones de manera correcta sin interferencias en la comunicación y dando el servicio adecuado a los usuarios de la información que necesitan; se comunica de manera eficiente a través del protocolo RS232 con el módulo GM862-GPS, así mismo funciona de manera correcta la comunicación con sus esclavos.

El trabajo de ejecución del programa realiza los siguientes pasos

Comunicación con el PIC maestro.

- Información mandada a la computadora por el puerto serial.
- Recibe la trama, la entiende y realiza la lógica difusa programada.
- Información de o las palabras mandadas al PIC y este las recibe.
- PIC se comunica dependiendo de la interpretación con el módulo, y pide la información interpretada o la envía en caso opuesto

La aplicación de los programas en este momento ha resultado exitosa



Figura 41. Desplegado de información del PIC en una LCD

Al momento de pedir información al pic este responde de forma adecuada procesando muy bien la información y emitiendo una respuesta certera. Para saber si se estaba ejecutando un código adecuado se implemento un despliegue de la instrucción para corroborar que así había sido. Figura 42



Figura 42. Interpretación adecuada del comando enviado

Una vez que se enviaba la instrucción del PIC a través del puerto serial era necesario al momento de recibirla procesarla de manera adecuada algo que ha ocurrido de manera favorable dando una adecuada interpretación. Figura 43



Figura 43. Respuesta emitida por el Modem y procesada

Al término de la instrucción ejecutada es necesario validar que la información ha sido correcta, pues en algunos casos la red GSM y sobre todo la red GPRS pueden no enviar la información de manera adecuada por alguna circunstancia no prevista, para poder validar esta información se ha desplegado una instrucción de lo que se ha ejecutado.



Figura 44. Envío de información a través de GSM/GPRS

El software creado para la computadora ha dado muy buenos resultados teniendo un buen comportamiento al momento de pedir datos, comunicación con el microcontrolador, al configurar los parámetros establecidos, y sobre todo la comunicación que se hace con el microcontrolador que a su vez se conecta con el modem.

A screenshot of the 'Infogene' software interface. The window title is 'Infogene'. The main content area is titled 'Información básica del producto:' and contains several input fields with their current values: 'SIM Insertada en Modem:' is 'Listo'; 'Calidad de la señal:' is 'Buena'; 'Carga de batería:' is 'Cargada'; 'Identificación de marca:' is 'Telit'; 'Identificación de revisión:' is '357467030000194'; 'Identificación de movil internacional:' is '334020306797316'; 'Estado de actividad:' is 'Listo'; 'Fecha:' is '05/09/2011'; and 'Hora:' is '15:40:26'. At the bottom right, there is a button labeled 'Conseguir Datos'.

Figura 45. Configuración información básica

Se ha logrado configurar el modem a través de los distintos comandos enviados.



Figura 46. Configuración general



Figura 47. Configuración GPRS

Definir contexto PDP(Packet Data Protocol)

Número de conexión particular PDP: 1

Tipo PDP: IP

Acces Point Name: internet.itelco

Dirección PDP: 0.0.0.0

Configurar parámetros Parámetros sugeridos

Figura 48. Configuración PDP

Calidad mínima de servicio del perfil:

Identificación del contexto PDP: 1

Prioridad de clase: 0

Retraso de clase: 0

Fiabilidad de clase: 1

Rendimiento máximo de clase: 3

Rendimiento de clase: 0

Configurar parámetros Parámetros sugeridos

Figura 49. Configuración calidad mínima de servicio

Calidad requerida de servicio del perfil:

Identificación del contexto PDP: 1 ▼

Prioridad de clase: 0 ▼

Retraso de clase: 2 ▼

Fiabilidad de clase: 2 ▼

Rendimiento máximo de clase: 4 ▼

Rendimiento de clase: 2 ▼

Figura 50. Calidad requerida de servicio

configuracionGSM

Configuración GSM

Agregar número: 4421430385

Seleccionar servicio: Modo 2 ▼

Figura 51. Configuración GSM

Configurar Socket:

Identificación de la conexión del socket:

Identificar contexto PDP (Packet Data Protocol):

Tamaño del paquete a utilizar por TCP: bytes

Máximo tiempo de intercambio (1-65535):

Máximo tiempo de espera de conexión (10 - 1200):

Tiempo de envío de datos (1 - 255):

Figura 52. Configuración Socket

Abrir conexión de internet:

Identificar contexto de conexión del socket:

Seleccionar el protocolo de transmisión:

Puerto de conexión:

Dirección IP o dirección web:

Finalizar conexión TCP:

Conexión del puerto UDP:

Modo de conexión:

Figura 53. Configuración conexión a internet

Abrir conexión FTP:

Puerto del servidor:

Usuario:

Contraseña:

Modo:

Figura 54. Configuración FTP

Tipo de transferencia FTP:

Binaria

ASCII

Figura 55. Configuración transferencia FTP

En cuanto a los resultados del programa para el celular, ha respondido bien a una instalación correcta y al manejo de las características definidas anteriormente. Como se puede observar en la figura 56, se ha instalado en un celular marca SAMSUNG, modelo SG550 (TOCCO), la selección de cada una de las acciones requeridas funciona correctamente.

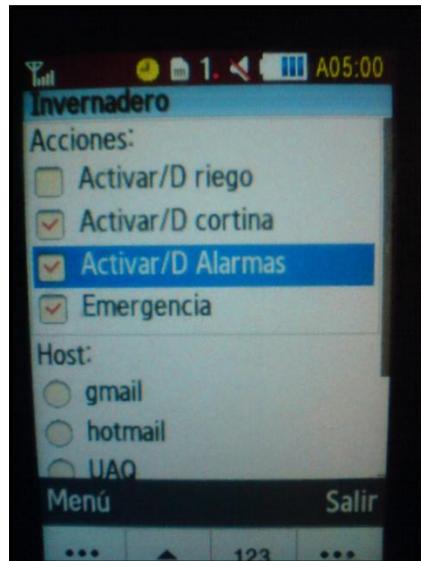


Figura 56. Programa instalado

Una operación importante a señalar es el haber configurado previamente el módulo TELIT con los datos correspondientes para que este en determinado tiempo pudiese funcionar como servidor, para ello se ha hecho uso del comando AT#FRWL en donde se tiene que configurar una ip y una máscara, para la investigación de ha configurado de la siguiente manera:

```
AT#FRWL=1,"197.158.1.1","255.255.0.0"
```

Otra operación importante para poder recibir una información es configurar el socket de conexión para que este pueda “escuchar”. En donde se requiere identificar la conexión, configurar el estado de socket a modo de “escuchar” y agregar un puerto. Por lo anterior tenemos la siguiente forma de configuración:

AT#SL = 1, 1, 6543

En el momento que los datos estén entrando se producirá una interrupción que tiene el siguiente comando:

SRING: 1

Donde se ha pedido al modem la introducción de datos en el socket numero 1, y para aceptar esta conexión se tiene que utilizar el comando AT#SA seguido del número de identificación al cual se requiere corresponder, como a continuación se presente:

AT#SA = 1

Posteriormente aparecerá un mensaje de conexión por parte del módulo y los datos serán recibidos.

Actualmente cabe señalar el control con el celular ha causado diversos problemas en cuestión de la programación del mismo, sin embargo la aplicación y la comunicación se realiza de buena forma en la red GSM.

A continuación se muestra en la figura 57 la selección del host modem TELIT el cual será identificado para enviar datos o pedir que este envíe datos al celular.

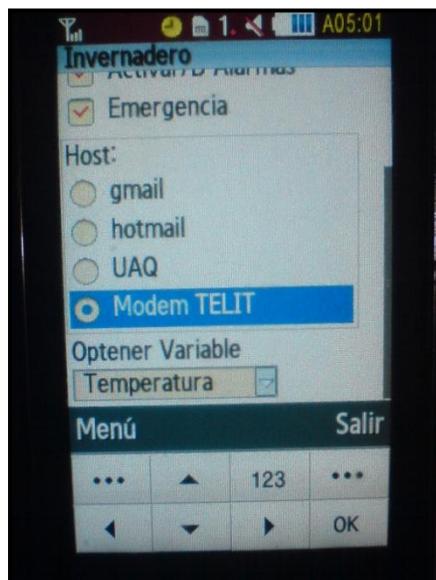


Figura 57. Selección de host

Es fundamental el monitorear de las variables pero se ha dejado a la elección del usuario saber que variable desea conocer, para que este a su vez pueda hacer uso de buenas decisiones y enviarlas a los servidores

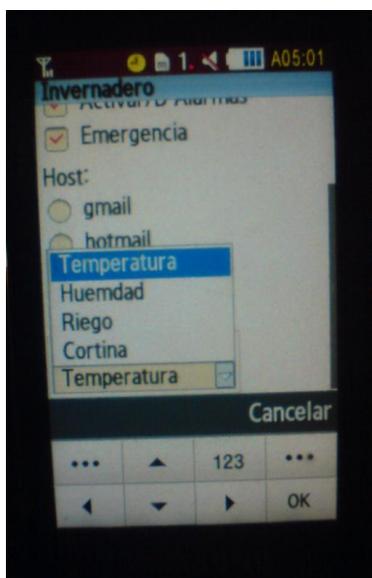


Figura 58. Selección de variable a monitorear

Fundamental la elección de la operación requerida, figura 59.



Figura 59. Selección operación a realizar.

4.2 Hardware

El equipo funciona correctamente, sin ningún problema mecánico o eléctrico, los tiempos de respuesta están regidos para esclavo dependiendo de la instrucción del maestro. Todo el equipo está dentro de su garantía otorgada por el fabricante en forma y tiempo. El manejo adecuado de cada uno de los instrumentos que conforman el producto total generará a corto plazo una retribución, en el rendimiento de los mismo así como el control adecuado del invernadero pues los instrumentos funcionan de manera correcta.

4.3 Conclusiones y desarrollos futuros

Se han cumplido satisfactoriamente los objetivos propuestos en tiempo y forma el proyecto.

La plataforma de comunicación basada en GPRS para la administración de datos en invernaderos automatizados ha involucrado varias áreas para poder realizar de manera correcta el control en el invernadero, sobre todo en el área de electrónica y programación pues fue el principal objetivo del trabajo además de incorporar tecnología necesaria a nuestro país pues los productores en el sector primario podrán verificar cual es el estado del invernadero y esto se refleja en automático en la calidad del producto así como el incremento del mismo. Las áreas de crecimiento para futuros proyectos son innumerables pues el uso de los celulares principalmente está teniendo un alto crecimiento, esto debido por las características que actualmente incorporan haciendo posible el instalar aplicaciones desarrolladas para un uso en particular.

El proyecto ha sido concluido satisfactoriamente con lo propuesto al inicio en la hipótesis y objetivos, demostrando que el control sobre el invernadero puede ser más eficiente si se lleva una herramienta móvil consigo la cual permite tomar las mejores decisiones y alertar si algo no está funcionando de acuerdo a lo esperado, pero durante su desarrollo se ha visto que se pueden añadir más características sobre lo ya construido en el software y hardware.

LISTA DE ACRÓNIMOS

ARFCN	Absolute Radio Frequency Channel Number
AT	Attention command
BA	BCCH Allocation
BCCH	Broadcast Control Channel
CA	Cell Allocation
CBM	Cell Broadcast Message
CBS	Cell Broadcast Service
CCM	Current Call Meter
CLIR	Calling Line Identification Restriction
CTS	Clear To Send
CUG	Closed User Group
DCD	Data Carrier Detect
DCE	Data Communication Equipment
DCS	Digital Cellular System
DGPS	Differential GPS, the use of GPS measurements, which are differentially corrected
DNS	Domain Name System
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTMF	Dual Tone Multi Frequency
DTR	Data Terminal Ready
GGA	GPS Fix data

GL	Geographic Position – Latitude/Longitude
L	
GLONASS	Global positioning system maintained by the Russian Space Forces
GMT	Greenwich Mean Time
GNSS	Any single or combined satellite navigation system (GPS, GLONASS and combined GPS/GLONASS)
GPRS	Global Packet Radio Service
GPS	Global Positioning System
GSA	GPS DOP and Active satellites
GSM	Global System Mobile
GSV	GPS satellites in view
HDLC	High Level Data Link Control
HDOP	Horizontal Dilution of Precision
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IRA	International Reference Alphabet
IWF	Interworking Function
MO	Mobile Originated
MT	<i>either</i> Mobile Terminated <i>or</i> Mobile Terminal
NMEA	National Marine Electronics Association
NVM	Non Volatile Memory
PCS	Personal Communication Service
PDP	Packet Data Protocol
PDU	Packet Data Unit

PIN	Personal Identification Number
PPP	Point to Point Protocol
PUK	Pin Unblocking Code
RLP	Radio Link Protocol
RMC	Recommended minimum Specific data
RTS	Request To Send
SAP	SIM Access Profile
SCA	Service Center Address
SMS	Short Message Service
SMSC	Short Message Service Center
SMTP	Simple Mail Transport Protocol
TA	Terminal Adapter
TCP	Transmission Control Protocol
TE	Terminal Equipment
UDP	User Datagram Protocol
USSD	Unstructured Supplementary Service Data
UTC	Coordinated Universal Time
VDOP	Vertical dilution of precision
VTG	Course over ground and ground speed

LISTA DE COMANDOS

AT+CPIN
AT+CGMI
AT+CGMM
AT+CGMR
AT+CGSN
AT+CIMI
AT+CPAS
AT+WS46
AT+CBST
AT+CMEE
AT+CREG
AT+COPS
AT+CSQ
AT+CCLK
AT+CBC
AT+CGCLASS
AT+CGATT
AT+CGREG
AT+CGDCONT
AT+CGQMIN
AT+CGQREQ
AT+CGACT
AT#SGACT
AT#SD
AT#FTOPEN
AT#FTPTYPE
AT#FTPPUT
AT#SH
AT#ESMTP

AT#EUSER
AT#EPASSW
AT#EADDR

LITERATURA

- [1] Equipo Editorial Explorando México, “Campo Mexicano”, 2010.

- [2] © 2006 MaxStream, Inc., Confidential & Proprietary, 802.15.4 and ZigBee™, Manual de documentación.

- [3] José María Romero López, “GPRS: ¿Una tecnología de transmisión?”, Antenas de Telecomunicaciones. Diciembre del 2002.

- [4] Radio-Electronic Resources and analysis for electronics engineering, Cellular Telecoms, GPRS (General Packet Radio Service).

- [5] Advanced PIC Microcontroller Projects with the PIC18F Series Dogan Ibrahim.

- [6] TCP/IP Embedded Internet Applications, Edward Insam PhD, BSc.

- [7] 3GPP TS 23.060, General Packet Radio Service.

- [8] Digi Connect, “Cellular IP Connections”, 2011.

- [9] K. Sklower, “The PPP Multilink Protocol”, Agosto 1996.

- [10] W. Richard Stevens, Bill Fenner, Andrew Mrudolff, “UNIX Network Programming: The sockets networking API”, Pearson Education Inc, Third edition, 2004.

- [11] Kenneth Ingham, “Una Historia y estudio de Firewalls de Red”, 2002.

- [12] J. Postel, J. Reynolds, ISI, “File Transfer Protocol (FTP)”, Octubre 1985.

- [13] Mariana del Rey, “Transmission Control Protocol”, Instituto de las Ciencias de la información, Septiembre de 1981.

- [14] S. Deering, Cisco, R. Hinden, Nokia, “Internet Protocol”, The Internet Society, 1998.

[15] Mariana del Rey, "Internet Protocol", Instituto de las Ciencias de la Información, Enero 1980.

[16] Sun Microsystem, "Arquitectura de Aplicaciones", Junio 2009.

[17] i-SAFE America Inc, "Peer to Peer Networking", 2008.

[18] European Telecommunications Standards, "GSM", Mayo 2011.

[19] Paul Correia, "Guía práctica del GPS", Marcobo S. A., 2002.

[20] Muijzenberg, Erwin van den Banco Mundial "Una historia de invernaderos", Instituto de Ingeniería Agrícola de Wageningen, Holanda, 1980.

ANEXOS

PROGRAMA PIC:

```
#include <18F4550.h>
#include <string.h>
#include <stdlib.h>
#uses XT,NOPROTECT,NOWDT
#use delay(clock=4000000)
#use rs232(rcv=PIN_C7,xmit=PIN_C6,baud=9600,bits=8,parity=N)
#define LCD_DATA_PORT getenv("SFR:PORTD")
#include <lcd.c>
#define buf_length 128

int buf[buf_length],str[buf_length],done[buf_length],go[buf_length],x[buf_length],y[buf_length];
int i=0,str_flag=0, t=0, s=0,j=0, n=0, g=0, b=0, c=0, f=0,l=0,m=0;
int a=0,e=0,coma=0,h=0,k=0,bn=0,cn=0,activar=0;
float noreset=0,west=0;
char platica;

int atg[9]={'$','G','P','S','A','C','P',':',0};
int gps[11]={'A','T','$','G','P','S','A','C','P',0x0D,0x0A};
int smse[22]={'A','T','+','C','M','G','S','=',' ','4','4','2','1','4','3','0','3','8','8',' ',' ','0x0D,0x0A};
int smsr[2]={'>',0};
int at[5]={'A','T',0x0D,0x0A,0};
int ok[4]={'O','K',0x0D,0};
int sigue[3]={1,0,0};
int ctrz[1]={0x1A};
int comillas[1]={'"'};

void descargar();
void analisis();
void enviar_sms();
void enviar_gps();

#int_rda
void rcv_data()
{
    buf[i]=getchar();

    if(buf[i] == 0x0A)
    {
        i=0;
        t++;
    }
    else if(buf[i] == 0x0D)
    {
        s++;

        if(s == 3)
        {
            a++;
            strcpy(str,buf);
            f=1;
        }
    }
}
```

```

    i=0;
}

for(j=0; j<= 127; j++)
{
    buf[j]=0;
}
i=0;
j=0;
}

else if(buf[i]== '>')
{
    bn=1;
    for(j=0; j<= 127; j++)
    {
        buf[j]=0;
    }
    i=0;
    j=0;
    buf[0]='>';
    buf[1]=0;
    strcpy(go,buf);
    buf[0]=0;
    buf[1]=0;
}

else
    i++;
}

void get_string()
{
    while(str_flag==0)
    regresa:
    if(str_flag==0)
    {
        goto regresa;
    }
    str_flag=0;
}

void main()
{
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    setup_adc(ADC_OFF);
    setup_adc_ports (NO_ANALOGS);
    lcd_init();

    While (TRUE)
    {
        lcd_putc("\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc, "SISTEMA GPRS UAQ");
    }
}

```

```

lcd_gotoxy(1,2);
printf(lcd_putc," FAC. ING.%i",bn);

if(f == 1)
    analisis();

//if(s==5)
//descargar();

delay_ms(100);

//ENVÍA CARACTERES AL MODEM TELIT

if(input(PIN_B0) == 1)
    enviar_gps();

delay_ms(100);

if(activar == 1)
    enviar_sms();

delay_ms(100);

//SE COMPARAN RESPUESTAS RECIBIDAS

if(strcmp(atg,go)==0)
{
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Posicionando GPS");

    for(b=1;b<=15;b++)
    {
        lcd_gotoxy(b,2);
        printf(lcd_putc,"|");
        delay_ms(200);
    }

    //delay_ms(2000);

    for(b=0;b<=127;b++)
    {
        if(str[b]==',')
            coma++;

        if(coma == 1 && m == 1)
        {
            if(str[b] != 'N')
            {
                y[h]= str[b];
                h++;
            }
        }
    }
    if(coma == 1)
        m=1;

    if(coma == 2 && l == 1)

```

```

        {
            if(str[b] != 'W')
            {
                x[k]= str[b];
                k++;
            }
        }
        if(coma == 2)
            l=1;
    }
    norest=(double)(atof(y));
    west=(double)(atof(x));
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    printf(lcd_putc,"X= %4.4f",west);
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Y= %4.4f",norest);
    activar=1;
    delay_ms(3000);
    descargar();
}

if(strcmp(smsr,go)==0)
{
    n++;
    lcd_putc('\f');
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Escribiendo GPRS...");
    lcd_gotoxy(1,2);
    printf(lcd_putc,"GPS");
    delay_ms(100);
    printf("Hola version 5N%4.4f W%4.4f--",norest,west);
    delay_ms(100);
    printf("%s",ctrz);
    lcd_gotoxy(1,2);
    printf(lcd_putc,"%s",ctrz);
    delay_ms(2000);
    descargar();
}
}

void descargar()
{
    for(c=0;c<=126;c++)
    {
        str[c]=0;
        buf[c]=0;
        done[c]=0;
        go[c]=0;
        x[c]=0;
        y[c]=0;
    }

    s=0;
    i=0;
    h=0;
    k=0;
    coma=0;
}

```

```

l=0;
m=0;
//activar=0;
}

void analisis()
{
  int u;

  f=0;
  for(u=0;u<=127;u++)
  {
    done[u]=str[u];
    if(done[u]!=':')
    {
      e++;
      done[u+1]=0;
      strcpy(go,done);
    }
  }
}

void enviar_sms()
{
  lcd_putc("\f");
  activar=0;
  for(g=0;g<=21;g++)
  {
    delay_us(350);
    //+lcd_putc("\f");
    //+lcd_gotoxy(1,1);
    printf("%c",smse[g]);
    printf(lcd_putc,"%c",smse[g]);
    delay_ms(300);
  }
}

void enviar_gps()
{
  lcd_putc("\f");
  for(g=0;g<=10;g++)
  {
    delay_us(350);
    //+lcd_putc("\f");
    //+lcd_gotoxy(1,1);
    printf("%c",gps[g]);
    //+printf(lcd_putc,"%c",smse[g]);
    //delay_ms(300);
  }
}

```

PROGRAMA PC:

//Ventana principal

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Drawing.Drawing2D;

namespace Robot_Delta
{
    public partial class Form1 : Form
    {
        private String sim;
        private String señal;
        private String bateria;
        private String marca;
        private String revisión;
        private String movilinternacional;
        private String edoactividad;
        private String fecha;

        private Object clasegprs;
        private Object adjuntar;
        private Object edored;

        private String t1,t2,t3;
        private String px, py, pz;
        private Double tt1, tt2, tt3;
        private Double tt12, tt22, tt32;
        private Double tt13, tt23, tt33;
        private Double t11, t12, t13;
        private Double ppx, ppy, ppz;
        private Double cx, cy, cz;
        private Double cx2, cy2, cz2;
        private Double cx3, cy3, cz3;
        private Double k,k1, k2;
        private Double jx1, jy1, jz1;
        private Double jx2, jy2, jz2;
        private Double jx3, jy3, jz3;
        private Double jq1, jq2, jq3;
        private Double jqq1, jqq2, jqq3;
        private Double vp1, vp2, vp3;
        private Double teta1, teta2, teta3;

        private Pen myPen;

        private String p;
        private String r;
```

```

private String o;
private String b;
private String ha;
private String v1;
private String v2;
private String v3;
private String cuenta;

private float hh;
private float rr;
private float o1= 0;
private double o2= (120 * Math.PI)/180;
private double o3 = (240 * Math.PI) / 180;
private float bb;
private float aa;
private float cuentas;
private float vv1;
private float vv2;
private float vv3;

public Form1()
{
    InitializeComponent();
}

private void rbdirecta_CheckedChanged(object sender, EventArgs e)
{
    label2.Visible = true;
    label4.Visible = true;
    label5.Visible = true;
    label6.Visible = true;
    tbt1.Visible = true;
    tbt2.Visible = true;
    tbt3.Visible = true;

    //label3.Visible = false;
    //label7.Visible = false;
    //label8.Visible = false;
    //label9.Visible = false;
    //tbx.Visible = false;
    //tby.Visible = false;
    //tbz.Visible = false;
}

private void rbinversa_CheckedChanged(object sender, EventArgs e)
{
    label2.Visible = false;
    label4.Visible = false;
    label5.Visible = false;
    label6.Visible = false;
    tbt1.Visible = false;
    tbt2.Visible = false;
    tbt3.Visible = false;

    // label3.Visible = true;
    //label7.Visible = true;
    //label8.Visible = true;
}

```

```

        //label9.Visible = true;
        //tbx.Visible = true;
        //tby.Visible = true;
        //tbz.Visible = true;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Configuracion conf = new Configuracion();
        DialogResult resulta = conf.ShowDialog();

        if (resulta == DialogResult.OK)
        {
            p=conf.H;
            r=conf.R;
            o = conf.O;
            b=conf.B;
            ha=conf.HA;
            cuenta = conf.Cuentas;

            hh = float.Parse(p);
            rr = float.Parse(r);
            bb = float.Parse(b);
            aa = float.Parse(ha);
            //cuentas = float.Parse(cuenta);
        }
        conf.Dispose();
        conf = null;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        foreach (String s in SerialPort.GetPortNames())
        {
            cbpuertos.Items.Add(s);
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (rbdirecta.Checked == true)
        {
            t1 = tbt1.Text;
            t2 = tbt2.Text;
            t3 = tbt3.Text;
            tt1 = float.Parse(t1);
            tt2 = float.Parse(t2);
            tt3 = float.Parse(t3);

            cx = (bb * Math.Sin(tt3)*Math.Cos(tt1+tt2)+aa*Math.Cos(tt1)-
            bb*Math.Cos(tt3)*Math.Sin(o1)-hh+rr)/(Math.Cos(o1)*Math.Pow(Math.Sin(o1),2));
            cy=(bb*Math.Cos(tt3)+cx*Math.Sin(o1))/(Math.Cos(o1));
            cz = bb * Math.Sin(tt3) * Math.Sin(tt1 + tt2) + aa * Math.Sin(tt1);
        }
    }

```

```

txtr.Clear();
txtr.Text = "Seleccionado Cinemática directa" + "\r\n\r\nResultado:
"
    + "\r\nCx= " + cx + "\r\nCy= " + cy + "\r\nCz= " + cz;
}
if (rbinversa.Checked == true)
{
    //px = tbx.Text;
    //py = tby.Text;
    //pz = tbz.Text;
    v1 = tbv1.Text;
    v2 = tbv2.Text;
    //v3 = tbv3.Text;

    ppx = double.Parse(px);
    ppy = double.Parse(py);
    ppz = double.Parse(pz);
    vv1 = float.Parse(v1);
    vv2 = float.Parse(v2);
    vv3 = float.Parse(v3);

    float ytr;

    double s1;
    double c1;
    double st3;
    double aux3;
    double g1;

    cx=ppx*Math.Cos(o1)+ppy*Math.Sin(o1)+hh-rr;
    cy=-ppx*Math.Sin(o1)+ppy*Math.Cos(o1);
    cz = ppz;

    s1= Math.Sin(o1);
    c1 = Math.Cos(o1);

    tt3 =Math.Acos((( -ppx * s1)+(ppy * c1))/ (bb));
    st3 = Math.Sin(tt3);

    g1=(2 * aa * bb * st3);
    k = (Math.Pow(cx, 2) + Math.Pow(cy, 2) + Math.Pow(cz, 2) -
Math.Pow(aa, 2) - Math.Pow(bb, 2))/ g1;
    tt2 = Math.Acos(k);

    k1=bb*Math.Sin(tt3)*Math.Cos(tt2);
    k2=bb*Math.Sin(tt3)*Math.Sin(tt2);

    tt1 = Math.Asin((k2*cx-(aa+k1)*cz)/(-Math.Pow(k2,2)-
Math.Pow(aa+k1,2)));
    t11 = (tt1*180)/Math.PI;

```

```

cx2 = ppx * Math.Cos(o2) + ppy * Math.Sin(o2) + hh - rr;
cy2 = -ppx * Math.Sin(o2) + ppy * Math.Cos(o2);
cz2 = ppz;

s1 = Math.Sin(o2);
c1 = Math.Cos(o2);

tt32 = Math.Acos((( -ppx * s1) + (ppy * c1)) / (bb));
st3 = Math.Sin(tt32);

g1 = (2 * aa * bb * st3);
k = (Math.Pow(cx2, 2) + Math.Pow(cy2, 2) + Math.Pow(cz2, 2) -
Math.Pow(aa, 2) - Math.Pow(bb, 2)) / g1;
tt22 = Math.Acos(k);

k1 = bb * Math.Sin(tt32) * Math.Cos(tt22);
k2 = bb * Math.Sin(tt32) * Math.Sin(tt22);

tt12 = Math.Asin((k2 * cx2 - (aa + k1) * cz2) / (-Math.Pow(k2, 2) -
Math.Pow(aa + k1, 2)));
t12 = (tt12 * 180) / Math.PI;

cx3 = ppx * Math.Cos(o3) + ppy * Math.Sin(o3) + hh - rr;
cy3 = -ppx * Math.Sin(o3) + ppy * Math.Cos(o3);
cz3 = ppz;

s1 = Math.Sin(o3);
c1 = Math.Cos(o3);

tt33 = Math.Acos((( -ppx * s1) + (ppy * c1)) / (bb));
st3 = Math.Sin(tt33);

g1 = (2 * aa * bb * st3);
k = (Math.Pow(cx3, 2) + Math.Pow(cy3, 2) + Math.Pow(cz3, 2) -
Math.Pow(aa, 2) - Math.Pow(bb, 2)) / g1;
tt23 = Math.Acos(k);

k1 = bb * Math.Sin(tt33) * Math.Cos(tt23);
k2 = bb * Math.Sin(tt33) * Math.Sin(tt23);

tt13 = Math.Asin((k2 * cx3 - (aa + k1) * cz3) / (-Math.Pow(k2, 2) -
Math.Pow(aa + k1, 2)));
t13 = (tt13 * 180) / Math.PI;

jx1 = Math.Cos(tt1 + tt2) * Math.Sin(tt3) * Math.Cos(o1) -
Math.Cos(tt3) * Math.Sin(o1);
jy1 = Math.Cos(tt1 + tt2) * Math.Sin(tt3) * Math.Sin(o1) +
Math.Cos(tt3) * Math.Cos(o1);
jz1 = Math.Sin(tt1 + tt2) * Math.Sin(tt3);

jx2 = Math.Cos(tt12 + tt22) * Math.Sin(tt32) * Math.Cos(o2) -
Math.Cos(tt32) * Math.Sin(o2);
jy2 = Math.Cos(tt12 + tt22) * Math.Sin(tt32) * Math.Sin(o2) +
Math.Cos(tt32) * Math.Cos(o2);

```

```

        jz2 = Math.Sin(tt12 + tt22) * Math.Sin(tt32);

        jx3 = Math.Cos(tt13 + tt23) * Math.Sin(tt33) * Math.Cos(o3) -
Math.Cos(tt33) * Math.Sin(o3);
        jy3 = Math.Cos(tt13 + tt23) * Math.Sin(tt33) * Math.Sin(o3) +
Math.Cos(tt33) * Math.Cos(o3);
        jz3 = Math.Sin(tt13 + tt23) * Math.Sin(tt33);

        // JQ MOTOR

        jqq1 = aa / (Math.Sin(tt2) * Math.Sin(tt3));
        jqq2 = aa / (Math.Sin(tt22) * Math.Sin(tt32));
        jqq3 = aa / (Math.Sin(tt23) * Math.Sin(tt33));

        jq1 = (jx1 + jy1 + jz1) * jqq1;
        jq2 = (jx2 + jy2 + jz2) * jqq2;
        jq3 = (jx3 + jy3 + jz3) * jqq3;

        vp1 = vv1 * Math.Cos(o1) + vv2 * Math.Sin(o1);
        vp2 = -vv1 * Math.Sin(o1) + vv2 * Math.Cos(o1);
        vp3 = vv3;

        teta1 = (jq1 * vp1) + (jq2 * vp2) + (jq3 * vp3);
        teta2 = jq2 * vp1 + jq2 * vp2 + jq2 * vp3;
        teta3 = jq3 * vp1 + jq3 * vp2 + jq3 * vp3;

        txtr.Clear();
        txtr.Text = "Seleccionado Cinemática Inversa \r\nθ11=" + "" + t11 +
"\r\n"
            + "θ12= " + t12 + "\r\n" + "θ13= " + t13;
        // tbo1.Text = " " + teta1;
        // tbo2.Text = "" + jq1 + "" + jq2 + "" + jq3;
        //tbo3.Text = "" + jx3 + "" + jy3 + "" + jz3;
    }
}

private void button4_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)//Si está abierto
    {
        serialPort1.Close();//Cerrar puerto

        //Actualizar botón
        btnabrir.Text = "Abrir Puerto";

        //Actualizar etiqueta
        label12.Text = "Puerto COM " + cbpuertos.SelectedItem.ToString() +
"cerrado";
    }
    else//Si está cerrado
    {
        //Primero configuramos las propiedades del puerto
        serialPort1.PortName = cbpuertos.SelectedItem.ToString();

        //Ahora podemos abrirlo
        serialPort1.Open();

        //Actualizar btnabrir

```

```

        btnabrir.Text = "Cerrar Puerto";
    }
}

private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    //Da tiempo para que lleguen todos los datos al puerto
    System.Threading.Thread.Sleep(100);

    //Leer datos
    String datos = serialPort1.ReadExisting();

    //Cambiar la propiedad Text de txtentrada de manera segura
    tbre.Invoke(new EventHandler(delegate
    {
        tbre.AppendText(datos + "");
    }));
}

private void button3_Click(object sender, EventArgs e)
{
    serialPort1.WriteLine(txtr.Text);
}

private void button4_Click_1(object sender, EventArgs e)
{
    tbre.Clear();
    txtr.Clear();
}

private void Form1_MouseDown(object sender, MouseEventArgs e)
{
}

private void label27_Click(object sender, EventArgs e)
{
}

private void rbtnauto_CheckedChanged(object sender, EventArgs e)
{
    activarbombas.Enabled = false;
    activarcortinas.Enabled = false;
    btnbombas.Enabled = false;
    btncortinas.Enabled = false;
    txttempmax.Enabled = false;
    txttempmin.Enabled = false;
    txthummax.Enabled = false;
    txthummin.Enabled = false;
    labmaxtemp.Enabled = false;
    labmintemp.Enabled = false;
    labhummax.Enabled = false;
    labhummin.Enabled = false;
}

```

```

        labvariables.Enabled = false;
    }

    private void rbtnmanual_CheckedChanged(object sender, EventArgs e)
    {
        activarbombas.Enabled = true;
        activarcortinas.Enabled = true;
        btnbombas.Enabled = true;
        btncortinas.Enabled = true;
        txttempmax.Enabled = true;
        txttempmin.Enabled = true;
        txthummax.Enabled = true;
        txthummin.Enabled = true;
        labmaxtemp.Enabled = true;
        labmintemp.Enabled = true;
        labhummax.Enabled = true;
        labhummin.Enabled = true;
        labvariables.Enabled = true;
    }

    private void btninfo_Click(object sender, EventArgs e)
    {
        Infogene info = new Infogene();
        DialogResult resulta = info.ShowDialog();
        if (resulta == DialogResult.OK)
        {
            txtr.AppendText("AT+CPIN\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CSQ\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CBC\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CGMI\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CGMM\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CGMR\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CGSN\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CIMI\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CPAS\r\n");
            System.Threading.Thread.Sleep(100);
            txtr.AppendText("AT+CCLK\r\n");
            System.Threading.Thread.Sleep(100);
        }
        info.Dispose();
        info = null;
    }

    private void btnconfgen_Click(object sender, EventArgs e)
    {
        Configuraciongenerales confgene = new Configuraciongenerales();
        DialogResult resulta = confgene.ShowDialog();
    }

```

```

    }

    private void btnconfigprs_Click(object sender, EventArgs e)
    {
        ConfiguracionGPRS configprs = new ConfiguracionGPRS();
        DialogResult resulta = configprs.ShowDialog();
        if (resulta == DialogResult.OK)
        {
            clasegprs = configprs.Clasegprs;
            adjuntar = configprs.Adjuntar;
            edored = configprs.Edored;
        }
        configprs.Dispose();
        configprs = null;
    }

    private void btnconfigsm_Click(object sender, EventArgs e)
    {
        configuracionGSM configsm = new configuracionGSM();
        DialogResult resulta = configsm.ShowDialog();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        Enviardatosared enviarred = new Enviardatosared();
        DialogResult resulta = enviarred.ShowDialog();
    }
}
}
}

```

//Ventana de configuración GPRS

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Robot_Delta
{
    public partial class ConfiguracionGPRS : Form
    {
        private Object clasegprs;
        private Object adjuntargprs;
        private Object edored;
        private String b;
        private String ha;
        private String cuentas;

        public Object Clasegprs
        {
            get { return clasegprs; }
        }
    }
}

```

```

        set { clasegprs = value; }
    }

    public Object Adjuntar
    {
        get { return clasegprs; }
        set { clasegprs = value; }
    }

    public Object Edored
    {
        get { return clasegprs; }
        set { clasegprs = value; }
    }

    public ConfiguracionGPRS()
    {
        InitializeComponent();
    }

    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void label14_Click(object sender, EventArgs e)
    {

    }

    private void ConfiguracionGPRS_Load(object sender, EventArgs e)
    {

    }

    private void label23_Click(object sender, EventArgs e)
    {

    }

    private void comboBox14_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void label17_Click(object sender, EventArgs e)
    {

    }

    private void comboBox15_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void comboBox16_SelectedIndexChanged(object sender, EventArgs e)
    {

```

```

}

private void comboBox17_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void comboBox18_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void comboBox19_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void label18_Click(object sender, EventArgs e)
{
}

private void label19_Click(object sender, EventArgs e)
{
}

private void label20_Click(object sender, EventArgs e)
{
}

private void label21_Click(object sender, EventArgs e)
{
}

private void label22_Click(object sender, EventArgs e)
{
}

private void button4_Click(object sender, EventArgs e)
{
}

private void button6_Click(object sender, EventArgs e)
{
    comboBox1.SelectedIndex = 0;
    comboBox2.SelectedIndex = 0;
    comboBox3.SelectedIndex = 1;
}

private void button1_Click(object sender, EventArgs e)
{
    clasegprs = comboBox1.SelectedIndex;
    adjuntargprs = comboBox2.SelectedIndex;
}

```

```

        edored = comboBox3.SelectedIndex;
    }
}

```

PROGRAMA CELULAR:

```

package src;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;
import com.siemens.icm.io.ATCommand;
import com.siemens.icm.io.ATCommandFailedException;
import com.siemens.icm.io.ATCommandListener;

public class Conexion extends MIDlet
{
    ATCommand ATC;
    ATCommandListener LIS;
    int hemosRecibidoDatos=0;

    //Constructor de la clase
    public Conexion()
    {
        //Inicializamos la clase ATCommand y el Listener
        try
        {
            LIS = new ATListener();
            ATC=new ATCommand(false);
            ATC.addListener(LIS);
        }
        catch (IllegalStateException e){e.printStackTrace();}
        catch (ATCommandFailedException e) {e.printStackTrace();}

        //Iniciamos una conexión TCP
        iniciaConexion();
    }

    //Definimos nuestra clase ATListener
    class ATListener implements ATCommandListener
    {
        //Sólo controlaremos un URC en este ejemplo
        //y será para saber cuándo recibimos datos
        public void ATEvent(String Event)
        {
            //Sacamos el evento que se ha producido:
            System.out.println("Evento: "+ Event);

            //Y si el URC corresponde a que han llegado datos ...

```

```

    if (Event.indexOf("SISR")>0)
    {
        //Mandamos leer los datos del socket
        System.out.println("Leemos los datos TCP");
        leerSocket();
    }
}
public void RINGChanged(boolean Event) {}
public void DCDChanged(boolean Event) {}
public void DSRChanged(boolean Event) {}
public void CONNChanged(boolean Event) {}
}

protected void startApp() throws MIDletStateChangeException {}

protected void pauseApp() {}

protected void destroyApp(boolean arg0) throws
MIDletStateChangeException {}

//Método para conectarnos con el servidor remosto
public void iniciaConexion()
{
    //Nos conectaremos a Google
    String direccionIP = "216.239.59.147?";
    String puerto = "80?";
    String r="";

    System.out.println("Iniciando secuencia conexión GPRS");

    try
    {
        r=ATC.send("AT^SICS=1,conType,GPRS0\r");
        Thread.sleep(100);

        r=ATC.send("at^sics=1,user,CLIENTE\r");
        Thread.sleep(100);

        r=ATC.send("at^sics=1,passwd,AMENA\r");
        Thread.sleep(100);

        r=ATC.send("at^sics=1,apn,internet\r");
        Thread.sleep(100);

        r=ATC.send("AT^SICI=1\r");
        Thread.sleep(100);

        r=ATC.send("AT^SISS=1,svrType,socket\r");
        Thread.sleep(100);

        r=ATC.send("AT^SISS=1,conId,1\r");
        Thread.sleep(100);

        r=ATC.send("AT^SISS=1,address,socktcp://"+ direccionIP + ":" + puerto + "\r");
        Thread.sleep(100);

        r=ATC.send("AT^SISO=1\r");
        System.out.println(r);
        Thread.sleep(3000);
    }
}

```

```

System.out.println("Enviando datos ...");

//Enviamos una petición a Google de la página principal
escribirSocket("GET / HTTP/1.1\r\n\r\n");
System.out.println("Datos enviados");

//Nos quedamos esperando en un bucle a recibir algo vía TCP
System.out.println("Esperando recibir un paquete de datos TCP ...");
while (hemosRecibidoDatos==0);

//Cerramos el socket
r=ATC.send("AT^SISC=1\r");
Thread.sleep(100);

//Paramos el programa
notifyDestroyed();

}
catch (Exception e){System.out.println("Error.");}
}

//Método para leer los datos del socket
public void leerSocket()
{
    try
    {
        //Mandamos el comando AT para leer lo que hay en el socket
        ATC.send("AT^SISR=1,1500\r");
        Thread.sleep(1000);

        //Asociamos el stream de entrada de datos
        InputStream is = ATC.getDataInputStream();
        //Y si hay datos disponibles los leemos ...
        if (is.available() > -1)
        {
            //Creamos un buffer del tamaño de los datos recibidos
            byte[] buffer = new byte[is.available()];
            //Leemos los datos y los cargamos en buffer
            is.read(buffer, 0, is.available());
            //Mostramos los datos por la salida estandar ...
            System.out.println(new String(buffer));
            //Activamos el flag, de esa manera ya no nos quedamos esperando
            hemosRecibidoDatos=1;
        }
    }

    //Gestión de las exepciones
} catch (IllegalStateException e) {e.printStackTrace();}
} catch (IllegalArgumentException e) {e.printStackTrace();}
} catch (ATCommandFailedException e) {e.printStackTrace();}
} catch (InterruptedException e) {e.printStackTrace();}
} catch (IOException e) {e.printStackTrace();}

}

//Método para escribir datos en el socket
public void escribirSocket(String cadena)
{
    OutputStream os;

```

```

try
{
    System.out.println("Enviado: " + cadena);
    //Asociamos el stream de salida al de la clase ATCommand
    os = ATC.getDataOutputStream();
    //Enviamos el comando AT^SISW de envío
    ATC.send("AT^SISW=1," + cadena.length() + ",0,0\r");
    Thread.sleep(500);
    //Escribimos los datos por el socket
    os.write(cadena.getBytes());
    Thread.sleep(500);
}
//Gestión de las excepciones
catch (IOException e) {e.printStackTrace();}
catch (IllegalStateException e) {e.printStackTrace();}
catch (IllegalArgumentException e) {e.printStackTrace();}
catch (ATCommandFailedException e) {e.printStackTrace();}
catch (InterruptedException e) {e.printStackTrace();}

}
}

```